Mixed-Language Arabic- English Information Retrieval

by

Mohammed Mustafa Ali

Supervised by: Dr. Hussein Suleman

Thesis is presented for the degree of Doctor of Philosophy

In the Department of Computer Science

University of Cape Town

. February 2013 .

I know the meaning of plagiarism and declare that all of the work in the document, save for which is properly acknowledged, is my own.

Mohammed Mustafa

DEDICATION

To my parents,
to my wife,
to my kids and
to my supervisor

ACKNOWLEDGMENTS

My special gratitude goes to Dr. Hussein Suleman, my supervisor, who was very helpful and a considerable advisor. He spares no effort maintaining my ideas and views to this research. I owe him a huge thank you for his tireless recommendations and encouragement and for his gentle personal guidance. He was patient and cooperative - without him, I would never have got to the completion of this thesis. He was always helpful, honest and inspiring in criticism. He was much more than just the supervisor - he inspired me with his way of doing research.

I want to acknowledge also Prof. Douglas Oard at the University of Maryland, College Park, for his keen guidance and support. I have especially benefitted from his comments and from different discussion sessions with him. He spares no time answering my doubts and inquiries. I am also thankful to him for organizing a colloquium on this research at the Computational Linguistics and Information Processing (CLIP) laboratory, Institute for Advanced Computer Studies (UMIACS) at the University of Maryland, College Park - USA. I really feel indebted to Prof Doug.

I would like also to appreciate the members of the CLIP laboratory at the University of Maryland, for their priceless comments at the colloquium. I especially convey my special thanks to Prof. James Mayfield at the Johns Hopkins University and Prof. Philip Resnik at the University of Maryland, College Park for their valuable comments and advice.

I would like to acknowledge and thank for his contribution my colleague Dr. Mohammed Elshazali at the Department of Astronomy, University of Cape Town for his countless hours of discussions and suggestions. I owe him much for his support.

My thanks go to all people at the Academy of Arabic Language, Sudan Office for allowing me to have a deep look at the type of difficulties that usually occur when translating/transliterating Arabic scientific terms. I owe them much for organizing a workshop on the current challenges of Arabic information retrieval, in general and on the motivation for this thesis, in particular.

My thanks also go to the staff members, PhD. students, M.Sc. students and undergraduate students at the Sudan University of Science and Technology and University of Khartoum, for their contribution in building a test collection for this thesis. Special thanks also go to Prof. Izzedin Mohammed Osman for his encouragement and for his support during the journey of this thesis. He was very supportive and I very indebted to him.

For their support and emotional encouragement, my thanks to my close friends Mr. Mohammed Osman and Miss. Rihab Eltayeb. They were always there strengthening my resolve when needed.

Undoubtedly, my thanks go to my small family at the Digital Libraries (DL) laboratory in the department for their regular discussions. I am especially grateful to Morwan for his help during the final stages of this thesis.

I also acknowledge the financial assistance of the University of Cape Town, the Telkom/NSN/Telesciences THRIP Centre of Excellence and National Research Fund (NRF) at South Africa. They sponsored much of this research work.

I am gratefully indebted to my family. My heartfelt thanks go especially to my parents for their endless support in my entire life. They were always there encouraging me in everything. I owe them very much that could not be repaid by just words.

Finally, very special thanks for my wife, Afag, who helped me in every way she could over the years of this thesis. She was always listening when I needed and sharing my thoughts and inspirations. I owe her much.

ABSTRACT

The Web is essentially cross-lingual and/or multilingual. It contains a large number of resources that are in a multitude of languages. However, diversity of languages on the Web happens not only with multiple versions in different languages but, also, in a single page with different languages together. This is especially true for non-English languages in which text switching/mixing (e.g. between Arabic and English) is very prevalent, especially in the scientific domain, due to the fact that most technical terms are borrowed from English and/or they are neither included in the native (non-English) languages nor have a precise translation/transliteration in these native languages. This makes it difficult to search only in a non-English (native) language because either non-English-speaking users, such as Arabic speakers, are not able to express terminology in their native languages or the concepts need to be expanded using context. This results in mixed queries and documents in the non-English speaking world. For example, 'ماذا نعني بال Asymmetric key' (meaning: what is meant by Asymmetric key) is a mixed query, written in both English and Arabic languages. In that perspective, it may no longer be possible to constrain non-English speakers to single languages in searching.

Current search engines and traditional CLIR (and MLIR) systems, which allow users to retrieve documents in a language that is different from the query language, did not handle mixed-language querying adequately and did not exploit this natural human tendency. This is because, in most cases, their weighting algorithms, indexing methods and ranking approaches are strongly optimized for monolingual queries. Even if the queries are translated, the mixed-language feature is handled as if the query or/and document is written in a monolingual language and, thus, the majority of the techniques results in a biased result list towards mixed documents. Due to this biasing, monolingual relevant documents even if they are highly relevant, would be ranked at lower levels and, thus, they could be easily missed by users. Additionally, Since terms in mixed documents - mostly non-English documents in scientific domains - are often accompanied by their translations, e.g., deadlock الإقفال, such co-occurring terms often raise the scores of mixed documents in which they occur and, thus, cause them to earn additional weights that are not part of their original scores.

This thesis attempts to address the problem of mixed querying in CLIR. It proposes mixed-language (language-aware) approaches in which mixed queries are used to retrieve most relevant documents, regardless of their languages. To achieve this goal, however, it is essential firstly to suppress the impact of most problems that are caused by the mixed-language feature in both queries and documents and which result in biasing the final ranked list. Therefore, a cross-lingual re-weighting model was developed. In this cross-lingual model, term frequency, document frequency and document length components in mixed queries are estimated and adjusted, regardless of languages, while at the same time the model considers the unique mixed-language features in queries and documents, such as co-occurring terms in two different languages.

Furthermore, in mixed queries, non-technical terms (mostly those in non-English language) would likely overweight and skew the impact of those technical terms (mostly those in English) due to high document frequencies (and thus low weights) of the latter terms in their corresponding collection (mostly the

English collection). Such phenomenon is caused by the dominance of the English language in scientific domains. Accordingly, this thesis also proposes reasonable re-weighted Inverse Document Frequency (IDF) so as to moderate the effect of overweighted terms in mixed queries. The re-weighted IDF is computed by combining document frequencies of terms with weights, particularly down-scaling factors, computed from their corresponding sub-collections. Estimation of sub-collection weights is based on an assumption that a sub-collection with a higher number of documents is expected to be more useful and have more significance.

These cross-lingual re-weighting approaches can be used when a single index is employed for indexing both mixed and monolingual documents or when documents, regardless of the language(s), are placed into a traditional distributed architecture. In particular, in the latter approach, besides the cross-lingual re-weighting model, a new indexing architecture that suits the mixed-language feature in documents was also proposed. The architecture minimizes major drawbacks of the two indexing approaches while have their advantages with respect to mixed-language queries and documents. For the purpose of conducting experiments and evaluating these proposed approaches, a new multilingual and mixed Arabic-English corpus on the computer science domain was created and statistically tested. This is because the majority of current test collections are monolingual or collected from the news genre.

Test results showed that the proposed cross-lingual re-weighting model, whether it is used with a centralized index or a conventional distributed architecture, could yield statistically significant better results, with respect to mixed-language queries, when it is compared to traditional approaches in CLIR and MLIR. Results also revealed that re-estimating weights for co-occurring terms in mixed documents could result in better performance. The use of the proposed indexing architecture with cross-lingual re-weighting in a traditional distributed architecture showed also that the approach is beneficial to mixed-language IR systems.

The results of the developed approaches prove to be efficient for improving retrieval via mixed-language querying, resulting a comparable effectiveness to monolingual retrieval using English queries.

Although the focus of this research is on the Arabic and English languages, the methodology proposed can be easily adapted to any language pairs, making it a practical solution especially in scientific domains in non-English languages. The study has shown that language-aware IR systems are important in the non-English speaking world and the proposed solution could better serve such non-English speaking users, who are always unable to approximate their ideas to search engines and/or need to retrieve most relevant documents rather than exact matching of terms between queries and documents, regardless of their languages.

Table of Contents

List of Tables

List of Figures

Original Research Publications

This thesis partially consists of the following papers:

1. Mohammed Mustafa, Izzedin Osman and Hussein Suleman, "Indexing and Weighting of Multilingual and Mixed Documents", in Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technology (SAICSIT), Cape Town, South Africa, ACM 52 (110-120) October 2011. The paper is a part of both chapter four "Mixed-Language Information Retrieval" and chapter six "Evaluation".
2. Mohammed Mustafa, Izzedin Osman and Hussein Suleman, "Building a Multilingual and Mixed Documents Corpus", in Proceedings of the Arabic Language Technology International Conference (ALTIC), Alexandria, Egypt, October 2011. The paper is a summary for chapter five "Building the Test Collection".
3. Mohammed Mustafa and Hussein Suleman, "Multilingual Querying", in Proceedings of the Arabic Language Technology International Conference (ALTIC), Alexandria, Egypt, October 2011. The paper is a summary for a portion of both chapter one "Introdcution" and chapter four "Mixed-Language Information Retrieval"
4. Mohammed Mustafa, Hisham AbdAlla and Hussein Suleman, "Current Approaches in Arabic IR: A survey", in Proceedings of the 11th International Conference on Asian Digital Libraries (ICADL), Bali, Indonesia, December 2008, LNCS 5362, Springer Verlag, 2008. This is an outline for chapter 3 "Arabic Information Retrieval: Current-state-of-art"
5. Mohammed Mustafa and Hussein Suleman, "Language-Aware Multilingual Information Retrieval", in Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technology (SAICSIT) – M&D Symposium, Wilderness, South Africa, October 2008.

CHAPTER

1

Introduction

As more digital information is made available, the Web continues to become the foremost channel for communication and the largest data repository, in which information can move freely with no physical boundaries or international borders. Billions of pages are becoming available everyday and the number of searches exceeds 500 million per day (Zhang and Lin 2007). In spite of globalization of information resources, English is the most popular language. Statistics provided by Miniwatts Marketing Group (2013) reported that English is the most used language on the Web and more than one-fourth (26.8%) of Web users use English in May 2011.

Besides the large number of English speaking users, such dominance of English on the Web is caused also by the fact that several organizations create English versions of their websites (besides those in their native languages) and of their broad business needs, probably to be widely accessible. Governments around the world also imposed English as a formal language, to some extent, in their educational and governmental spheres, e.g. some Arabic countries, probably to confirm the maximum information acquisition and to increase the intake and reach of knowledge. As a result of such policies, English was, and still is, the most dominant language for scientific articles, lexicons, dissemination of information and different types of knowledge.

However, while English remains the most popular language, its share has been declining over the past few years. It is reported that English has one of the slower rates of growth (during the time period 2000-2011) among the top 10 languages on the Internet in May 2011. The percentage of Internet growth in terms of usage during this time period has dropped to 301.4% in May 2011 (Miniwatts Marketing Group, 2013). This is because English content on the Web has been challenged by other languages - Arabic and Chinese are examples. Such non-English languages are growing rapidly, in terms of both users and Web usage, and witness the explosive growth of non-English speaking users on the Web. Figure 1.1 shows Internet user growth in May 2011. For example, Arabic has the highest rate of growth (2,501.2%) on the

Web with regards to users (Miniwatts Marketing Group, 2013) and its contents are predicted to double every year (Chung et al., 2006). This growth of non-English languages on the Web is also because some governments enforce that national corporations and organizations publish some material like people's heritage, geographical data and educational technical material in native languages. Such languages include Arabic, Chinese and Russian. The same criterion also holds when it comes to formal native academic material, like references, books, etc, which merely are written in the native languages.

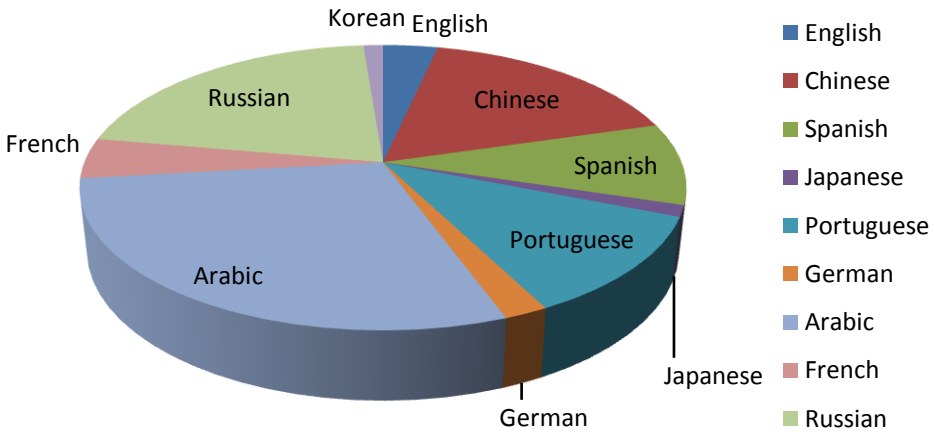


FIG. 1.1: The top 10 languages used on the Web in terms of growth during the time period 2000-2011 (Miniwatts Marketing Group, 2013).

Accordingly, more and more pages on the Web are written in different languages. This results in globalized information and a large number of resources that are very much diverse and in a multitude of languages. This feature makes the Web essentially cross-lingual and/or multilingual. But, this linguistic multiplicity and moving towards an international community should no longer be a barrier for accessing information, regardless of its language, on the Web. When users of any language need to search in any language for a particular topic in any language, the search results should no longer be limited to the native languages of those users or the language of their requests. For such users, Cross-Language Information Retrieval (CLIR) provides some solutions. In CLIR, users are able to obtain represented relevant information (known as document sets) in a monolingual language that is different from the language they used in their information need requests (known as queries) (Nie, 2010). For example, a user may type his/her query in Arabic, but a relevant document is written/retrieved in English or any other language. However, as the international community includes many languages, the CLIR task can be extended to be multilingual (this happens when information resources or document sets are in more than one language). This is the Multilingual Information Retrieval (MLIR) problem (Chen and Gey, 2004b), in which a user query is posted in a single monolingual language to document sets in more than one language, whereas results, after being merged, are presented in a single unified list in multiple languages. Both CLIR and MLIR approaches involve some type of automatic translation, in which both documents and query are unified into a single language, e.g., to translate query into documents' language(s) and, hence, the process reduces to a monolingual search and retrieval.

But, diversity of languages on the Web happens not only with multiple versions in different languages, as both CLIR and MLIR presume but, also, in a single page with different languages together in a tightly integrated text. This is especially true for non-English languages in which text switching (i.e. between both Arabic and English) is very prevalent and increasing every day. This mixed-language trend is not limited to the Web. In fact, its appearance on the Web is caused by its spread in everyday life in the non-English-speaking world, e.g., mixing languages together in talking. This habit of multilingualism is very common in multilingual communities, in which the bar between different multi-cultures is lowered and in which natives use more than one language in their daily business lives and everyday demands (such as teaching, economy, business, culture, sports, news and hobbies). In such communities, natives are able to express some keywords in languages other than their native tongue or vice versa. The following examples illustrate this natural human tendency in this non-English-speaking world:

- From personal experience, the typical Arabic speaker speaks a mixture of tightly-integrated words in both English and Arabic (and various slang variants).
- Europeans tend to be multilingual since they usually have a broad knowledge of several foreign languages other than their native ones (Sigurbjornsson, et al., 2005).
- Hong Kong speakers typically speak Cantonese with many English words (Chung, 2008).
- The Capetonian speaks English with many scattered Afrikaans words and/or local slang with English. This is a commonly known trend for Capetonians.
- Non-English scholars in different types of business, including science, are more likely to use mixed languages in their presentations.
- It is commonly known that Hindi is the one of the most widely spoken languages in India but speaking English on the fringes with Hindi is very common.
- Countries of the East Asian region are becoming multilingual in that users are familiar with the use of more than one language in business and ordinary life (Gey et al., 2005).
- Examples include also lectures where some text is best expressed in an indigenous/home/local language while other text may best be expressed in a variant of English. This mixed language grammar is emerging in everyday life (SMS, Facebook, etc).

This mixed-language trend is known as code-switching and has been one of the major focus areas for research in linguistics, sociology and psycholinguistic fields (Cheung and Fung, 2004), especially after the Web emerged. It is noticed, however, that in most of these bilingual/multilingual communities, the common factor in their mixed-language tendency is the use of English as a pivot/second language. In addition to historical backgrounds related to the early days of colonization in these countries, especially developing ones, the phenomenon of mixed-language use is being caused by the dominance of English as the most widely used language all over the world, as illustrated above.

But, with the growth of the Internet, especially in the few last decades, the mixed-language feature has begun to spread on the Web and gradually non-English natives, who are bilingual, begin to search the Web in a mixture of languages - mostly with English on the fringes but not at the core. They often do this in order to approximate their information needs more accurately, rather than using monolingual queries written in their native-tongue languages in searching. In their study of Web searching behaviour, Reih

and Reih, (2005) showed that some users may post queries in their native languages or a foreign language while others prefer to enter multilingual queries. Lu, et al. (2006) showed that the main reasons behind using mixed language querying in Web search are caused by the use of computer technologies and the fact that some Chinese words do not have a popular translation. The findings were concluded from 77,000 multilingual queries that were extracted from a query log of a search engine. Aula and Kellar (2009) found that users usually show different search strategies, including the use of multiple languages in searching, when searching for their information needs.

This new type of search can be identified as mixed or multilingual querying. It is also referred to as the bilingual query (Reih and Reih, 2005). A mixed query is a query written in more than one language – usually bilingual. For instance, the query 'مفهوم الـ polymorphism', (meaning: concept of polymorphism) is a mixed query that is expressed in two languages (Arabic and English). English portions in mixed queries are often the most significant keywords. Another example, 说明' integrity constraints', (meaning: explain integrity constraints) is a mixed query that is written in bilingual languages (Chinese and English).

In the same context, *a mixed or a multilingual document* can be defined as any document that is written in more than one language (Fung et al., 1999), even if it contains only one or more words. In such a document, there is a primary language and a secondary language, which is mostly English, as well, whose text is often presented/scattered in terms of terms/portions/snippets/phrases/paragraphs.

This mixed-language feature is widespread in non-English scientific domains, in which it is commonly known that most terminology is borrowed from the English language. Figure 1.2 on the next page shows an example of a Chinese mixed document taken from the Web[1]. Figure 1.3 shows another example of several parts acquired from different Arabic mixed documents and taken from the Web, in the computer science domain. In the latter figure, each part taken from a single document appears in a single colour.

Although the primary language in these documents are the non-English ones, the English parts are mostly significant terms and are expected to be good candidates for search, e.g., technical terms or proper nouns. It is also noted that mixed-language text in these documents are written into two different forms. The first form is written in a tightly-integrated or text-switching manner between the two languages (tightly-integrated portions between Arabic and English or Chinese and English). For instance, the tightly-integrated portions in Figure 1.3 are highlighted in green.

The second form of mixed-language text, which is the most common, consists of similar text (terms/phrases/snippets) description in both non-English and English languages. Probably in such a case, the scientific non-English term is accompanied by its corresponding translation in English so as to refine non-English terms. In Figure 1.3, co-occurring terms, in which English terms are introduced as translations to refine the Arabic terms, are presented in most documents. The Arabic words and their translations are highlighted in yellow. This feature of co-occurring terms in non-English documents is interesting and has been widely used. For example, Zhang and Vines (2004) stated that in Chinese Web pages English terms are very likely to be the translations of their immediately preceding Chinese terms

[1] web.nuu.edu.tw/~carlu/ncp_sc/FCh09.ppt

and, hence, the feature was intensively used in mining to extract translations of a great number of terms in queries.

作業系統的組成

- 作業系統指的是一種軟體，此種軟體由許多程式所組成
  - 程序管理器(process manager)
  - 記憶體管理器(memory manager)
  - 虛擬記憶體管理器(virtual memory manager)
  - 輔助記憶體管理器(secondary storage manager)
  - 檔案管理器(file manager)
  - 保護系統(protection system)
  - 命令翻譯系統command interpreter system)

「監督程式」(monitor)

作業系統類型

- 單人單工系統(Single-User Single-Tasking)
- 多工系統(multi-tasking system)
- 多程式系統(multi-programming system)
- 多重處理系統(multi-processing system)
- 分散式系統(distributed system)
- 多執行緒系統(multi-thread system)

- 分時系統(time sharing system)
- 及時系統(real time system)
- 整批處理系統(batch processing system)
- 交談式系統(interactive system)等

Running 狀態

- 在「running」狀態中的程序利用CPU執行時，有三種可能的情況會發生：
  - 正常或不正常結束：離開系統，進入「terminate」狀態
  - 時間配額(time quantum)用完：只有採用「巡迴型排程法」(round robin scheduling)才可能發生此情況。因分配給程序的時間配額用完，因此程序會再次回到「ready」狀態，等候分配下一次的CPU使用權。
  - 等候事件發生或等候輸出入裝置之使用權：程序將進入「waiting」狀態

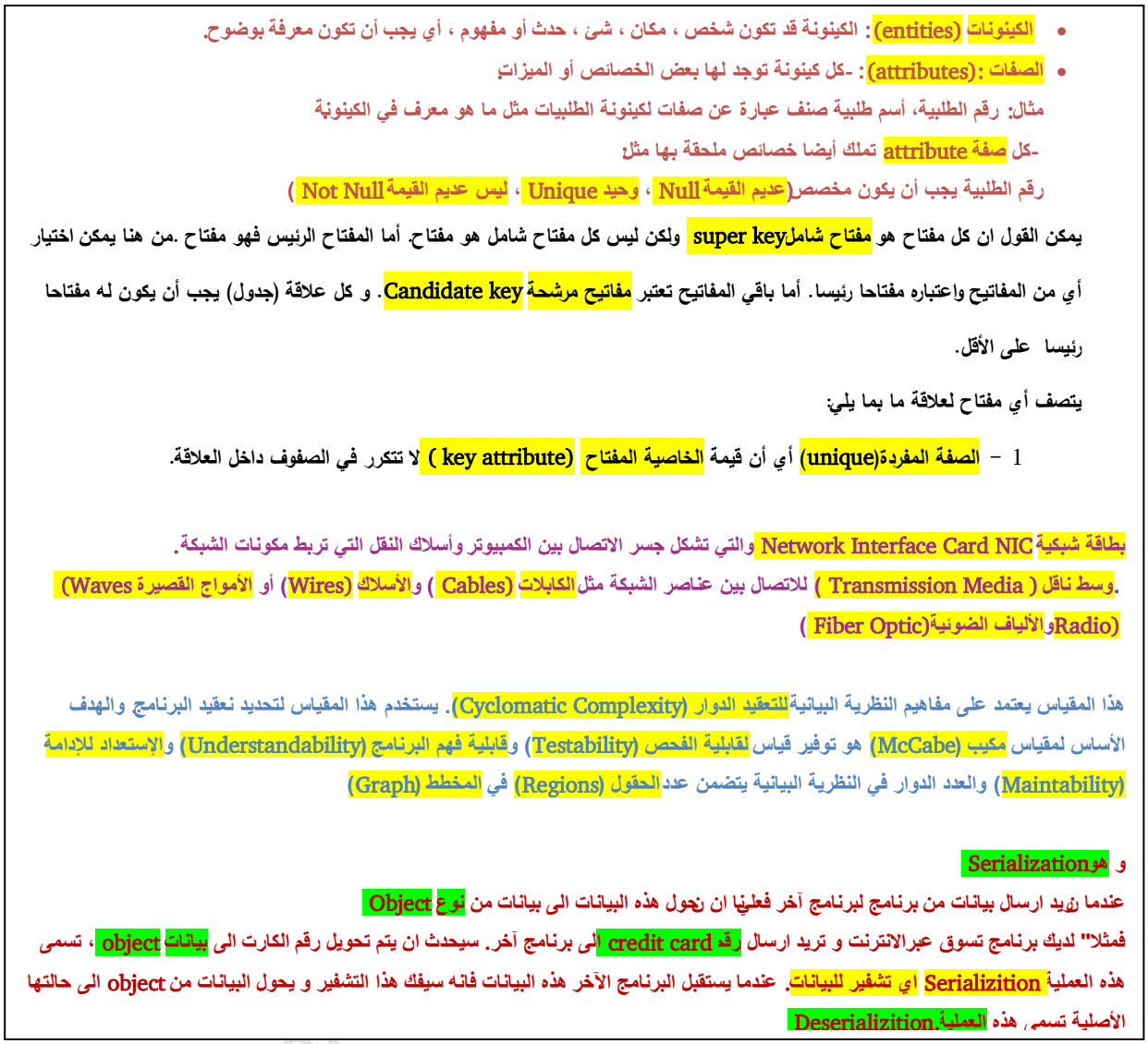
「waiting」狀態

- 在「waiting」狀態的程序則必在等候的事件已發生或已取得輸出入裝置的使用權後會進入「ready」狀態

本文切換(context switching)

- 由於程序可能無法在取得CPU使用權後便一次將整個程式執行完畢；最有可能的情況是必須分多次才能將程式執行完畢，因此系統必須進行「本文切換」(context switching)動作
- 「本文切換」是指作業系統儲存目前正在執行的程序的狀態並將下一個要執行程序之狀態載入系統並開始其執行的動作

FIG. 1.2: An Example for a mixed Chinese-English document taken from the Web.



- الكينونات (entities): الكينونة قد تكون شخص ، مكان ، شئ ، حدث أو مفهوم ، أي يجب أن تكون معرفة بوضوح.
- الصفات :(attributes): -كل كينونة توجد لها بعض الخصائص أو الميزات.

مثال: رقم الطلبية، أسم طلبية صنف عبارة عن صفات لكينونة الطلبيات مثل ما هو معرف في الكينونة

-كل صفة attribute تملك أيضا خصائص ملحقة بها مثل:

رقم الطلبية يجب أن يكون مخصص(عديم القيمة Null ، وحيد Unique ، ليس عديم القيمة Not Null)

يمكن القول ان كل مفتاح هو مفتاح شامل super key ولكن ليس كل مفتاح شامل هو مفتاح. أما المفتاح الرئيس فهو مفتاح .من هنا يمكن اختيار أي من المفاتيح واعتباره مفتاحا رئيسا. أما باقي المفاتيح تعتبر مفاتيح مرشحة Candidate key. و كل علاقة (جدول) يجب أن يكون له مفتاحا رئيسا على الأقل.

يتصف أي مفتاح لعلاقة ما بما يلي:

1 – الصفة المفردة(unique) أي أن قيمة الخاصية المفتاح (key attribute) لا تتكرر في الصفوف داخل العلاقة.

بطاقة شبكية Network Interface Card NIC والتي تشكل جسر الاتصال بين الكمبيوتر وأسلاك النقل التي تربط مكونات الشبكة.

وسط ناقل (Transmission Media) للاتصال بين عناصر الشبكة مثل الكابلات (Cables) والأسلاك (Wires) أو الأمواج القصيرة (Waves Radio)والألياف الضوئية(Fiber Optic)

هذا المقياس يعتمد على مفاهيم النظرية البيانيةللتعقيد الدوار (Cyclomatic Complexity). يستخدم هذا المقياس لتحديد تعقيد البرنامج والهدف الأساس لمقياس مكيب (McCabe) هو توفير قياس لقابلية الفحص (Testability) وقابلية فهم البرنامج (Understandability) والإستعداد للإدامة (Maintability) والعدد الدوار في النظرية البيانية يتضمن عددالحقول (Regions) في المخطط (Graph)

و هوSerialization

عندما نريد ارسال بيانات من برنامج لبرنامج آخر فعلينا ان نحول هذه البيانات الى بيانات من نوع Object

فمثلا" لديك برنامج تسوق عبرالانترنت و تريد ارسال رقم credit card الى برنامج آخر. سيحدث ان يتم تحويل رقم الكارت الى بيانات object ، تسمى هذه العملية Serializition اي تشفير للبيانات. عندما يستقبل البرنامج الآخر هذه البيانات فانه سيفك هذا التشفير و يحول البيانات من object الى حالتها الأصلية تسمى هذه العملية Deserializition.

FIG. 1.3: Several parts acquired from different mixed Arabic-English documents and taken from the Web.

Additionally, it is noted that sometimes the co-occurring terms are phrases, which are composed from two or more words. For instance, in Figure 1.3 the phrase المفتاح الشامل is accompanied with its English translation Super Key, composing the bilingual phrases 'Super Key المفتاح الشامل'. Sometimes the phrase consists of three words, e.g., 'Entity and Relationship Diagrams مخطط الكينونة والعلاقات'. It is also noted that the same term/word/phrase may be written in the same document in different and distant positions, sometimes even not in the same vicinity, but in multiple languages. For instance, in Arabic documents there are cases in which the scientific term "deadlock", for example, occurs in both Arabic and English, but in a single document.

Given these trends, this thesis attempts to explore the mixed-language feature in both documents and queries. In this chapter the problem is introduced. The following section, section 1.1, illustrates the major problems in existing search engines and current CLIR and MLIR approaches when the mixed-language

feature in both queries and documents is considered. Section 1.2 describes issues related to mixed querying, as it is the focus of this thesis, in Arabic IR. The section also provides reasons to use mixed Arabic-English querying in the Arabic scientific domain. The next section, section 1.3, reports briefly on different proposed approaches to handle mixed-language querying. The research questions are provided in section 1.4. Section 1.5 describes the contribution of the thesis.

1.1 Mixed-Language Querying and Documents: Problems

In a broad sense, a typical IR system, whose goal is to find relevant information for users according to their search requests, should carry out a matching process between searches (represented as queries) and information sources (represented as document sets). This matching would result in a list of documents scored according to their relevance to queries. These closely related issues are often handled within a ranked retrieval IR model. In CLIR, as a specialized problem in IR, the IR task includes an additional translation component that would be integrated as direct matching often would be unsuccessful due to differences of languages in document sets and queries. Therefore, most CLIR systems adopt a translation technique in order to satisfy the need for matching between queries and documents. Thus, either the query, the documents set or both is to be translated. Accordingly, the underlying assumption is that a typical CLIR reduces the matching process between documents and queries to a translation followed by a monolingual retrieval. In addition, it is often assumed that the document set(s) is presented in a single language (or in several monolingual languages), even if the set contains mixed documents, which are either deliberately/cautiously ignored or handled as if they are written in a monolingual language. This is noted in the large number of experiments in the literature. The same assumption of monolingualism is implicitly presumed also when it comes to MLIR, where the task is extended to multilingual retrieval.
This principal assumption makes most algorithms optimized for monolingual queries, even if these queries are translated, rather than for mixed queries. Examples include weighting algorithms, indexing methods and ranking approaches in existing CLIR and MLIR systems. Therefore, the majority of the current search engines and traditional CLIR systems perform poorly when handling mixed querying, because, in most cases, they fail to provide the most relevant documents, whose retrieval is one of the major goals in the IR process. This what the next section explores. In particular, the section identified major drawbacks in both existing search engines and CLIR and MLIR approaches, when the mixed-language feature in queries and documents is considered.

1.1.1 Experimenting with Mixed-Language Queries in Current Search Engines

Three major drawbacks can be identified, when mixed queries are posted to search engines.

1.1.1.1 Skewed Result Lists towards Mixed Documents

Whenever a mixed query is used, using existing search engines, the search result is often biased towards documents that exactly contain the same terms that are presented in the mixed query, regardless of its constituent languages. Figure 1.4 shows an example of a mixed Arabic-English query 'مقدمة في ال threading' (meaning: introduction to threading), submitted to the Google Web search engine[2]. The search was conducted in late December 2012. Appendix A shows another example for a Chinese-English query.

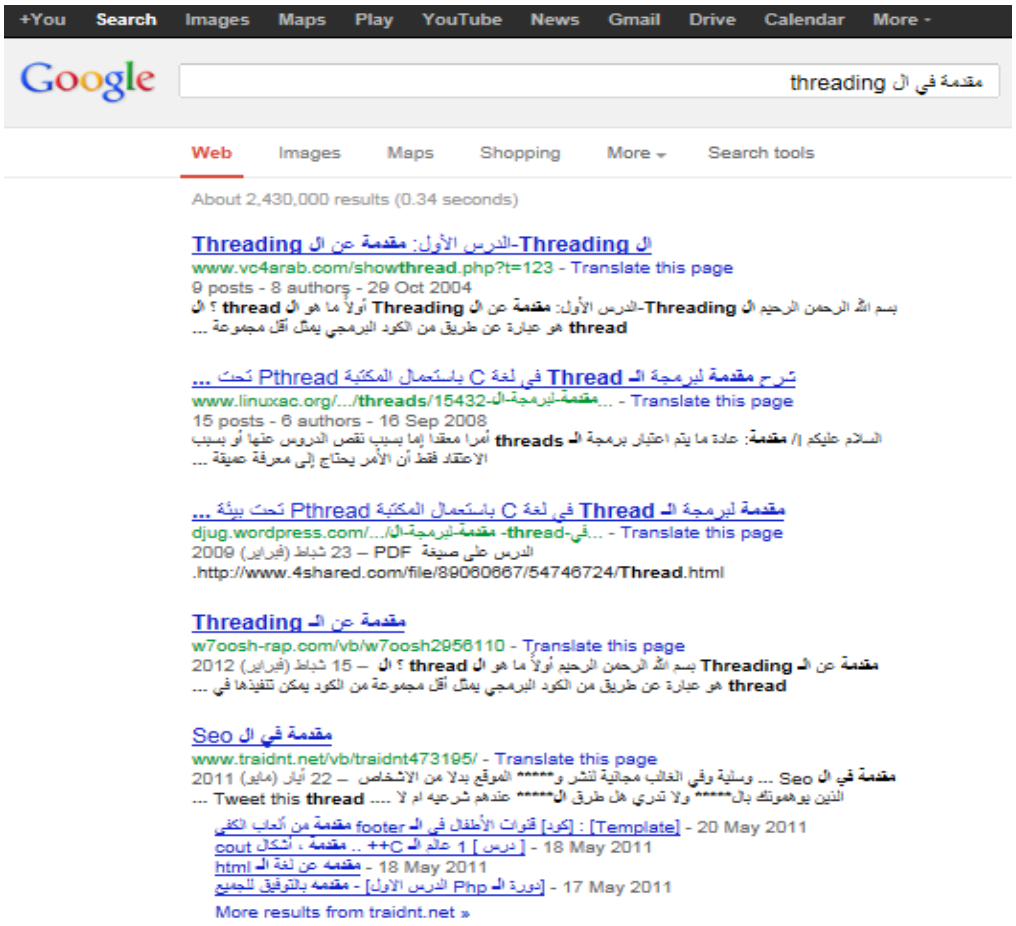


FIG. 1.4: An example of a mixed bilingual Arabic-English query submitted to Google.

In the retrieved result, the top of the ranked list, which is assumed to contain the most relevant documents, and the entire list as well, is dominated by mixed documents (with Arabic as a primary language), rather than the most relevant documents. Furthermore, the retrieved list, when it was investigated, revealed that there are many monolingual and highly relevant documents, which are mostly written in English, at lower ranks.

This bias of search results towards mixed documents is due to the retrieval process being based on monolingual weighting, in which terms are dealt with as if the mixed query is in a single language.

[2] http://www.google.com

Apparently such an assumption would likely make the top of the ranked list be dominated by those mixed documents, as it appears in the figure, when a mixed query is used for searching because in most cases they are the only documents that could have the majority of terms present in mixed queries, resulting in higher scores for these mixed documents. Such behviour biases retrieval as it is desirable to answer users' requests according to relevance of documents and independent of the language(s) of queries and documents, rather than just retrieving documents containing query terms, even if they are bilingual.

1.1.1.2 Uncompetitive Scores of Monolingual Documents

An additional undesirable impact of monolingual weighting, regardless of languages in mixed queries and documents, is that weighting of terms in the Arabic portion of multilingual queries is handled in a similar way to English term weighting. Such handling would result in monolingual document weightings, even if highly relevant, being no longer competitive to mixed documents' weightings. For example, in a mixed query like 'شرح ال polymorphism', (meaning: explain polymorphism) the scores of monolingual English documents will be computed from the word polymorphism only, whereas it would probably be calculated from the entire mixed query for mixed documents. Accordingly, many monolingual relevant documents, mostly written in English, would be ranked at the lower level of the retrieved list and could be easily missed by users, even if they are highly relevant. This is not a desirable trait as users often tend to examine only the top documents (i.e. 20) of search results (Jävelin and Kekäläinen, 2002).

1.1.1.3 Biased Result Lists towards less Important Terms

It was shown by many researchers, e.g. Fung, et al. (1999), Lin and Chen (2003), Lu, et al. (2006), Cheung and Fung (2005) and Aula and Kellar (2009), that English snippets in mixed queries are usually rich, significant and good candidates for search, whereas non-English portions, Arabic for example, mostly consist of general purpose vocabulary and/or stopwords. This is especially true in a technical domain. Accordingly, weights of non-technical terms in mixed queries (Arabic portion in mixed queries, for example) would likely distort the impact of the remaining matches, including those that are English technical. As the number of documents in English is expected usually to be much higher than any other non-English, this would probably have a significant impact on weighting of English terms because such terms would likely appear in many documents and, thus, there is low importance for the technical English terms. Contrarily to this scenario, non-English terms, for example Arabic ones, would probably result in higher importance and, thus, the ranked list would be biased towards general vocabulary terms, rather than English, which are mostly significant, e.g., technical terms or proper nouns. Note that the result list is already biased towards mixed documents as described in section 1.1.1.1.

1.1.2 Mixed-Language Problems in CLIR and MLIR Approaches

Neither mixed-language queries nor searches for mixed-language documents have yet been adequately studied. It was shown that the grounding belief is that the CLIR task is a translation followed by a monolingual retrieval, and, thus, most algorithms are strongly optimized for monolingual queries, rather than for mixed queries. Some CLIR studies investigated the use of a hybrid bi-directional approach that merges both document translations, from one direction, with query translation, from the other direction. Some studies tested this approach at word levels (Nie and Simard, 2001; Aljlayl, et al., 2002; Chen and Gey, 2004a; Wang and Oard, 2006), while others explored the hybrid approach at document levels by merging result scores that were obtained from each unary translation direction (McCarley, 1999). Most results showed that such a combination is very useful, but the queries set in them were essentially monolinguals with a grounding base that the test collection is monolingual (in a different language from query's language) and the major aim is to disambiguate translation, rather than handling the mixed-language feature in queries and documents. Accordingly, the mixed-language feature in both queries and documents, e.g., co-occurring terms in different languages, are either ignored and/or not handled adequately.

When the retrieval task becomes multilingual, in which several monolingual document sets in different languages are used, the problem of mixed-language queries and documents is extended to include indexing methods. Generally, two basic approaches in MLIR are utilized for indexing: the centralized architecture and the distributed architecture. The centralized architecture puts all documents, regardless of their languages, into a single index (Nie and Jin, 2003). Queries are translated into all the target (documents) languages and concatenated to form a single merged multilingual query, which is submitted to the single mixed collection. It is noted that in CLIR the use of a single index for indexing documents is the most widely used approach. The majority of the different editions of conferences in CLIR, e.g., Text Retrieval Conference (TREC), mostly used a single index that is composed of all the documents, which are assumed to be monolingual. In other words, the centralized architecture of indexing is implicitly assumed.

The second type of indexing approach in MLIT is the distributed architecture. The distributed architecture indexes documents, which are presumed to be in several monolingual languages, in each language separately (Chen and Gey, 2004b). Thus, the number of indices is equal to the number of languages presented in the multilingual collection. Queries are translated also to all the target languages of all the indices. Then, a monolingual retrieval in each index (language corresponding sub-collection) is carried out using its corresponding query. Next, all individual intermediate results are merged into a single ranked list.

Each of the two indexing approaches has some pros and cons with respect to mixed queries and documents. Some of these cons are severe, especially in the weighting components of mixed queries. The next sections discuss these limitations.

1.1.2.1 Mixed-Language Problems in a Centralized Index

The use of a centralized index has been shown to have a major drawback, which is overweighting (Lin and Chen, 2003). With respect to mixed-language problems, other limitations are also identified.

1.1.2.1.1 Overweighting Due to Incomparable Sizes of Collections

Overweighting means that weights of documents in small collections are often preferred (Lin and Chen, 2003). In particular, the number of documents increases, as all documents regardless of their languages are put in a single pool, while the number of occurrences of a term is kept unchanged and, thus, resulting in larger weights for terms that appear in small collections (the Arabic collection, for example). In this thesis, this type of overweighting is called *traditional overweighting*.

1.1.2.1.2 Biased Term Frequency

When it comes to mixed-language queries and documents, another major drawback is that weights of similar terms across languages are assigned and computed independently- as if they are different or in a single monolingual language (a merged query like 'Inheritance الوراثة ' in which each word is a translation of the other). Consequently, the same drawback of dominance of mixed documents on top of ranked retrieval lists would probably occur. Furthermore, the Term Frequency (TF) of terms that are cross-lingually similar within documents would likely be skewed towards the term with the highest term frequency, despite the fact that these terms (the source and its translation) are akin to each other, but cross-lingually. In this thesis, this drawback is called *biased term frequency*.

1.1.2.1.3 Biased Document Frequency and Overweighting

The same problem of cross-lingual terms that are computed individually affects also Document Frequency (DF) computation in that the latter would likely be counted individually for each term, although both terms are cross-lingually similar. This would likely skew the final list or the term may suppress the impact of its translation(s) (or vice versa). In that perspective, the term with low document frequency would likely overweight, even if its translation(s) is of low importance (high DF). This skew in DF is called in this thesis *biased document frequency,* whereas its consequent overweighting in terms is called *overweighting due to mixture of texts*.

1.1.2.1.4 Additional Weights for Co-occurring Terms

It was shown that in non-English documents, those in Arabic for example, terms are often accompanied by their translations, mostly in English. For example, the Arabic word الإقفال may be accompanied with its

English translation deadlock, resulting in the co-occurrence of (الإقفال deadlock) together but in two different languages. This is a very common feature in a considerable number of non-English documents, especially on the Web. Such co-occurrences of terms, however, when it comes to mixed-language queries in a centralized architecture would likely conspire to increase the scores of mixed documents and cause them to earn extra weights that are not really part of their weighting values. For example, consider the two documents that follow. The first document D_1 is a mixed document with Arabic as a primary language and English as a secondary language, whereas the second document D_2 is a monolingual English document:

D_1: " تؤدي عملية التطبيع normalization لإنشاء مجموعة جدول tables ذات..."

D_2: "The process of normalization leads to the creation of tables, whose…"

D_1 is the exact translation of D_2. However, since D_2 is in Arabic, as a primary language, the translated English term 'normalization' co-occurs with its Arabic equivalent 'التطبيع'. This is very common in non-English scientific writing, especially in references, e.g., Arabic references. In a centralized architecture, the big merged query may probably contain something like 'التطبيع normalization', as queries in such an architecture are usually concatenated. Such query would likely cause the mixed document D_1 to be ranked ahead of document D_2 because the Arabic term 'التطبيع' tends to co-occur with its equivalent English term in D_1. Thus, the document earns double weights, one for each term. This is a key problem in mixed-language queries and documents and it is called *distorted weighting*, in this thesis. Section 5.2 in the design chapter provides a complete illustrative example on the different types of drawbacks when a mixed-language trend occurs in both queries and documents.

1.1.2.2 Mixed-Language Problem in MLIR Distributed Architecture

When it comes to the traditional distributed architecture, the mixed-language problem in both mixed queries and documents will not be limited to weighting only, which was described above, but it is also extended to the question of how a mixed document is indexed in such an architecture. As was described earlier, the major approach in a traditional distributed architecture is based on distributing documents according to their languages. Thus, if a dispatching process is applied for the content of a mixed document (according to languages present in the mixed document), this would probably cause such a mixed document to lose its information richness and meaning, especially if its text is tightly-integrated among its constituent languages. Furthermore, partitioning such a mixed document into two or more sub-documents, depending on number of languages that are present in the document, causes the document to be *underweighted* in each sub-collection. This is a different behaviour when compared to the centralized indexing approach. While in the centralized index, mixed documents are favoured and monolingual documents, even if highly relevant, may be ranked at lower levels of result lists (the scores of mixed documents would be computed from the entire mixed and merged query), mixed documents in the traditional distributed architecture will be uncompetitive, even if highly relevant, and incomparable

to those monolingual ones in their corresponding sub-collections. This is because since the mixed document will be partitioned and indexed into several sub-collections (according to the languages that present in the document), its score will be computed from only a portion of the partitioned mixed document, rather than from the entire document and, thus, it will not be competitive to monolingual documents in their corresponding sub-collections.

1.2 Issues Related to Mixed Querying in Arabic IR

This thesis focuses on Arabic as a primary language, with English as a secondary language. The focus on these two languages is not surprising, as they are both among languages that have the largest economic plus commercial influence and both are of the six official languages of the United Nations (UN) (Chung, 2008).

With respect to mixed-language queries and documents, the Arabic language has some additional issues related to this mixed-language feature, specifically in the scientific domain. These are the regional variation problem and the reasons behind using Arabic-English mixed querying in Arabic IR.

1.2.1 Regional Variation Problem

Arabic documents that cover particular topics in technical domains are usually regionally variants. The problem of regional variation in Arabic, especially in scientific domains, is crucial. This is especially true when considering the Arabic-speaking world. The region has 22 countries[3], many of them with their own academy for the development of the language (Mirkin, 2010). Each academy translates/transliterates new terminology (referred to as Arabicization) individually, without a well-established coordination in most cases with its peers across the Arabic-speaking world (The Academy of Arabic Language, 2011). As a result, scientific modern terms in Arabic Gulf countries may be totally different from those in Levantine countries.

English Term	Arabic Term	English Term	Arabic Term
Hardware	العتاد	Hashing	التشتت
	المكونات المادية		البعثرة
	المكونات الفيزيائية		الفرم
Linked List	القائمة المتصلة	Symmetric key	المفتاح المتناظر
	القائمة المتسلسلة		
	اللائحة المترابطة		المفتاح المتماثل
	السلسلة المتصلة		

TABLE 1.1: Some regional variations in Arabic collected from the Web.

[3] Comoros is the 22nd Arabic country.

Table 1.1 shows some samples of these regional variations, gathered from the Web in the computer science domain. The significant proportion of Arabic technical terms on the Web are often found to be inconsistent and in different regional variants.

The problem of regional variants in scientific Arabic terminology grows dramatically with every new term added to the language. This problem affects mixed-language queries solely because in most cases it has an impact on translation of mixed queries; for example, many candidate translations would be produced. Furthermore, it makes the adoption of techniques like those that incorporate translation probabilities of candidate alternatives in weights of translations, an unwise decision because what appears as a superfluous translation in documents does not mean that this translation is deemed undesirable and documents in which it appears are irrelevant.

1.2.2 Why Arabic-English Mixed Querying/Writing

Arabic speakers, as many non-English speaking users, are often unable to accurately state terminology in their language, resulting in a mixed-language trend in speaking, writing and querying, especially in technical domains (The Academy of Arabic Language, 2011). When this mixed-language tendency is explored, five major reasons are identified.

1.2.2.1 Dominance of English

As the majority of credible content in the scientific domains on the WWW is available in English, most scientific terminology is borrowed from this popular language. As a result, it is not always possible for Arabic speakers to provide precise Arabic translations for newly added terms or/and not always feasible for those users to directly express their concepts in medicine and technology, for example. This makes it difficult to search in the native language, Arabic for example, because either the concepts need to be expanded or approximated using context.

A similar trend was also shown in Chinese. For example, in his analysis of reasons behind the use of Chinese-English mixed querying, Lu, et al. (2006) showed that one of the major causes for the phenomenon is that some Chinese words do not have a popular translation.

1.2.2.2 Irregular Translation/Transliteration of New Terminology

The translation/transliteration, if any, of newly added terms to Arabic (Arabicization), is not usually performed on a regular basis (The Academy of Arabic Language, 2011). This is a significant problem because it makes the Arabic language limited in its vocabulary of up-to-date terminology and, thus, Arabic speakers are unable to express some keywords in their native tongue and English technical terms are instead utilized.

1.2.2.3 Absence of Specialized Experts in Arabicization Process

One of the most significant problems with the Arabicization process, when it is performed, is that scientists who execute the process do not usually invite the experts and scientists in a given scientific domain to participate (The Academy of Arabic Language, 2011). This is a wide-spread problem in the Arabic world and it results in making translated/transliterated Arabic terms, in most cases, ambiguous. For instance, the Arabicization of the English terms: 'brainstorm', 'business re-engineering' and 'computerization / automation' are العصف الذهني ، الهندرة and الأتمتة, respectively (The Academy of Arabic Language, 2011). These Arabic words are ambiguous, chaotic and are almost not understood by Arabic speakers. Therefore, English scientific terms in the Arabic-speaking world, for example, are usually used to simplify ambiguous Arabic scientific terms.

1.2.2.4 Lack of Immediate Mirroring of New Terminology

Due to the lack of uniform workshops and seminars concerned with mirroring and reflecting newly translated/transliterated terms immediately to scientific domains, many Arabic speakers and users do not know the exact translations/meanings, even if they are found, for most terminology in scientific fields in their native languages. Thus, students at Arabic universities may ask a question like 'Deadlock ما هوالـ ', which is a tightly-integrated question that is presented in two languages and means 'what is deadlock' instead of 'ما هو الإستعصاء' because terms like deadlock are more meaningful and unambiguous to them.

1.2.2.5 Avoidance of Regional Variants

Although the English part of a multilingual query may have a proper translation in Arabic, which becomes popular after a relatively long period of time from the time of its translation/ transliteration, science scholars sometimes do not prefer to use such a proper translation in their communications or for searching across documents. This is because of the regional variation difficulty. Hence, in order to avoid missing some valuable documents due to its regional variation, Arabic users prefer to express terminology in English, rather than in Arabic, as most Arabic documents contain an Arabic regional variant term with its counterpart in English.

Beside these reasons, the nature of a certain specialized domain (i.e. computer science domain), whose vocabulary is merely based on the English language, plays also a role in using mixed queries. For example, if a student needs to search about the use of the mathematical function (exp), the OSI Model or PHP tutorials, he/she would probably use a mixed query, unless only English documents are desired.

1.3 Proposed Approaches

Given the above trends and the need to bring information to users in developing countries, this thesis introduces a mixed-language IR system to allow users to issue queries in a multilingual form to search across mixed and multilingual collections (a mixed collection usually contains mixed documents along with monolingual documents, whereas a multilingual collection consists of several monolingual documents in different languages). The key goal of the study is that the results of the user search should ultimately produce the most relevant documents regardless of the dominant language in the query words or documents or query language composition and regardless of the user's ability to express concepts in a particular language. To achieve this aim, the thesis proposes different algorithms that could handle the unique characteristics of the mixed-language problem in both queries and documents. In particular, two different sets of approaches are proposed. The first set was developed when a unified single index is used for indexing documents. This exactly suits the CLIR task and centralized indexing of MLIR. The second set of proposed approaches was developed when a traditional distributed architecture is used. The sections that follow describe these approaches.

For testing these approaches, however, a new mixed and multilingual test collection, with a mixed query set and relevance judgments, was created. This constructed corpus, which has been evaluated and validated using statistical tests, is specialized on common computer science and bilingual in both Arabic and English. This is because most currently available ad-hoc test collections, and almost CLIR collections, are either focused upon general-domain news stories, monolingual or consist of several monolingual corpora. Furthermore, their query sets are essentially monolingual and/or mixed documents in them are handled as if they are in a single language. Additionally, specialized corpora cover only few languages; Arabic is not among them.

It should be noted, however, that this thesis does not aim to develop a complete mixed-language IR system as such process involves many components, which are related to natural language processing and IR, that are not the focus of this thesis. Instead, the main goal is to develop and evaluate techniques which can handle mixed-language feature in both queries and documents and which can be easily incorporated in current IR systems.

1.3.1 Mixed-Languages Techniques in a Unified Index

The centralized architecture of MLIR is the closest approach to the mixed-language querying problem introduced in this thesis (mixed-language IR system). This similarity is because in both approaches the document collection is often multilingual in various languages (probably with some mixed documents). Additionally, both of the approaches utilize mixed queries. Recall that queries in traditional centralized architecture are often merged together to form a mixed merged query.

Therefore, when a single index is used, the key idea of mixed-language IR systems, which is described in detail in chapter 4, is focused on re-weighting. The proposed re-weighting, however, is of two different

aspects that can be applied either independently or in a combination. These aspects are cross-lingual re-weighting and weighted inverse document frequency.

1.3.1.1 Cross-Lingual re-weighting Model

The solution model of re-weighting is a cross-lingual re-weighing scheme that is developed to curb the impact of biased TF and biased DF. Within this cross-lingual weight, the effect of the distorted weights problem, in which terms earn extra weights due to co-occurring terms in different languages, is also suppressed. Thus, whenever a mixed query is posted, term frequency, document frequency and document length components are re-estimated according to this proposed re-weighting, which is an extended variant model of Structured Query Translation, proposed by Pirkola (Pirkola, 1998), but it is capable of handling the mixed-language feature in queries and documents.

1.3.1.2 Re-weighted Inverse Document Frequency

The proposed re-weighted Inverse Document Frequency (IDF) aims to suppress the effect of overweighting whether a collection is multilingual only or it is both mixed and multilingual. This is can be done by utilizing a reasonably re-weighted IDF of terms in mixed queries. The re-weighted IDF is computed by combining document frequencies of terms with weights, particularly down-scaling factors, for their corresponding sub-collections in the whole centralized collection. Estimation of sub-collection weights is based on an assumption that a sub-collection with a higher number of documents is expected to be more useful and have more significance.

1.3.1.3 Combined Cross-lingual and Weighted IDF approach

The proposed cross-lingual re-weighting could handle problems like biased term frequency, but terms, especially those non-technical in mixed and merged queries are still over-weighted. The re-weighted inverse document frequency handles overweighting in general. Thus, the combined cross-lingual and weighted IDF approach joins the two techniques together in order to moderate impact of most problems. This is achieved by applying the two techniques sequentially, starting with the proposed cross-lingual re-weighting model.

1.3.2 Mixed-Language Techniques in Traditional Distributed Architectures

Inspired by the question of how mixed documents can be indexed in the traditional distributed architecture, besides how they can be weighted, this thesis also explores the usefulness of developing a new MLIR architecture (indexing) that could avoid the problematic behaviours, which were described above, in both the distributed and the centralized architectures (indices). In particular, a new indexing

architecture that suits the distinguishing characteristic of mixed documents is proposed. This is achieved by combining the advantages of the centralized and distributed architectures for MLIR, while trying to moderate their drawbacks. A similar cross-lingual weighting to that proposed for the centralized architecture, but in terms of a probabilistic framework, is also utilized in this new architecture.

However, it should be noted that this thesis did not attempt to solve the problem of mixed-language IR in terms of distributed information retrieval (DIR) in which users would like to be able to simultaneously search different remote collections and in which problems like insufficient bandwidth, source representation, source selection and result merging are vital to its performance effectiveness. This is beyond the scope of the current work.

1.4 Research Questions

The primary goal of this research is to develop a set of algorithms for IR to handle mixed queries in multilingual and mixed corpora (language-aware/mixed-language IR system). In particular, the following questions are the core of this thesis:

1. What are the limitations of neglecting mixed-language features, in both documents and queries, on retrieval performance in current CLIR and MLIR approaches, whenever all documents are placed into a single index?
2. Whenever using the proposed cross-lingual weighting with a centralized mixed and multilingual index, what are the impacts on retrieval effectiveness of mixed querying? Is the retrieval effectiveness comparable to monolingual performance?
3. Is the co-occurrence of Arabic technical terms with their English equivalents could have significant impact on retrieved list by mixed queries?
4. How can traditional overweighting be moderated when using mixed-queries and what are the effects of re-weighting IDF of terms in mixed queries?
5. What is the impact of using the proposed hybrid architecture for indexing in a combination with the cross-lingual re-weighting for mixed queries? Is such an architecture efficient for indexing documents whether they are mixed or monolingual? And which is more effective to build a mixed-language IR system - is it the centralized approach or traditional distributed approach?

1.5 Contribution

The main contributions of this thesis can be summed up as follows:

1. Strong arguments for the mixed-language tendency and its importance in non-English countries, especially those that are still developing, are provided. This will facilitate understanding of the current status of the Web and the information searching needs of non-English users, who often need to approximate or expand their concepts to search engines. The arguments also provide useful guidelines for future search engines, in which it is essential to allow multilingual users to retrieve relevant information created by other multilingual users. This is especially true with the

information globalization on the Web. Furthermore, this can help in other fields such as machine translation systems, building lexicons and parallel/comparable corpora and transliteration.

2. New experimental techniques that aim to improve mixed-language IR systems are developed. The techniques consider the unique characteristics of mixed queries and documents, with special focus on the co-occurrences of terms in different languages, which are handled carefully by re-weighting terms while at the same time the impact of the overweighted terms is suppressed. This is done with the use of a centralized indexing approach and a distributed approach. Experiment results showed that such techniques for mixed-language IR systems could result in significant improvement in performance and could achieve comparable results to monolingual performance.
3. A novel architecture for indexing mixed documents in the traditional distributed architecture is also proposed by combining both the centralized and the distributed architectures, while re-weighting mixed documents whenever they are present.
4. A new Arabic-English test collection on common computer science vocabulary has been built and statistically tested. Beside its purpose to serve as a test-bed for research in IR, such a corpus would also provide a good opportunity for linguistic researchers to study the features of scientific Arabic in common computer vocabulary. This would help a lot in the future process of transliteration/translation of scientific terms and would help also in avoiding current obstacles in such processes. In fact, such a project has already been established on the basis of this work with one of the academies of the Arabic language.

1.6 Thesis Organization

The remainder of this thesis is organized as follows. **Chapter 2** gives an overview of CLIR. The most important components and different techniques are covered. MLIR approaches are also described. The state-of-the-art test collections and the literature on bilingual querying as well are also surveyed in this chapter. **Chapter 3** is an in-depth coverage of Arabic information retrieval. It reviews both monolingual and cross-lingual approaches that were proposed for the Arabic language. It also introduces the Arabic language and its characteristics, which make this language more challenging for the IR task and the impact of these features on IR systems.

The next chapter, **chapter 4**, introduces the proposed techniques for mixed-language IR systems. The first part of the chapter describes the techniques in term of a centralized architecture. It shows how the cross-lingual weighting of mixed queries and documents is estimated. Additionally, it presents how to reasonably re-weight overweighted terms. The second part of the chapter is devoted to the developed model of mixed-language IR in terms of a traditional distributed architecture. This typically includes an architecture for indexing and weighting of mixed documents and monolingual ones, as well. **Chapter 5** describes how the test collection was collected and how it was statistically tested. It shows also the needs for such a mixed and multilingual corpus in scientific domain. In **Chapter 6** the experiments conducted to

evaluate the developed techniques are presented. It contains also different comparisons for effectiveness of experiments. The last chapter, **chapter 7**, concludes the thesis with limitations. **Chapter 8** describes possible directions that could be considered for future work.

CHAPTER

2

Cross-Language Information Retrieval

The availability of information in different languages has resulted in a new type of information searching, known as Cross-Language Information Retrieval (CLIR), in which users search their information needs in a language that is different from language of information sources and, thus, the language barrier is crossed.

Before the explosive growth of the Web, however, research in Information Retrieval (IR) was focused on English and with English in mind for a long time. As this is no longer the case in the Web, research has begun to include many other languages. The first CLIR experiment was conducted in 1971 (**Salton, 1971**). During that time the work was focused upon library needs and on European languages. However, work in CLIR has been taken seriously on the mid of the 1990th and with the emergent of the Web. Accordingly, the first official cross-lingual run in the last decades was conducted in 1997 (Voorhees and Harman, 2000) using European languages. From that time, many languages have been explored and several conferences, institutions and governmental organization have begun to push the idea of cross-lingual access to information but with an eye on the multilingual nature of the Web. For example, TREC, sponsored by the NIST (National Institute of Standards and Technology), is one of the major evaluation workshops, which supports CLIR research in several languages. Other examples include the Cross-Language Evaluation Forum (CLEF), which is focused on the European languages, and the NII Test Collection for Information Research Systems (NTCIR), which is sponsored by the Japanese National Institute of Informatics (NII).

This chapter is concerned with the current approaches of CLIR. In section 2.1 an introduction to IR is provided. Major concepts and models behind IR are also outlined in this section. Section 2.2 introduces the major processes when a cross-lingual retrieval process is to be conducted. Through different sub-sections, the section also surveys different approaches that are proposed to cross the language barrier.

Following this, section 2.3 discusses proposed techniques in traditional multilingual information retrieval, in which the task of crossing language becomes multilingual, instead of bilingual. In section 2.4, the state-of-the-art of standard evaluation corpora is shown. The section presents different measures for evaluation as well. Section 2.5 introduces some related works. In particular, it discusses the issue of bilingual queries in library science and community. Finally, the chapter is summarized in section 2.6.

2.1 Information Retrieval

Information retrieval (IR) is the problem of satisfying users' information needs from unstructured data (like text, sound, image, etc), often known as a *collection*. In the context of textual IR, the major task of an IR system is to represent, store and manage information on the unstructured collections (referred to as set of *documents* in textual IR case) and provide a user with topical information on his information need (also referred to as a *query*) through an accessing mechanism to that collection. Defined in this way, the IR system should be able to: represent documents, which are often presented in a natural language, in somewhat searchable representations; represent queries, often a few words; and find documents that match the query representation with documents' representations.

2.1.1 Essential Processes in Information Retrieval

In the context of the definition above, the IR task can be decomposed into three main processes. These are: a searchable document representation over which the retrieval process is performed, a representation of a user information need and a matching process between the two representations, which results in a set of retrieved documents.

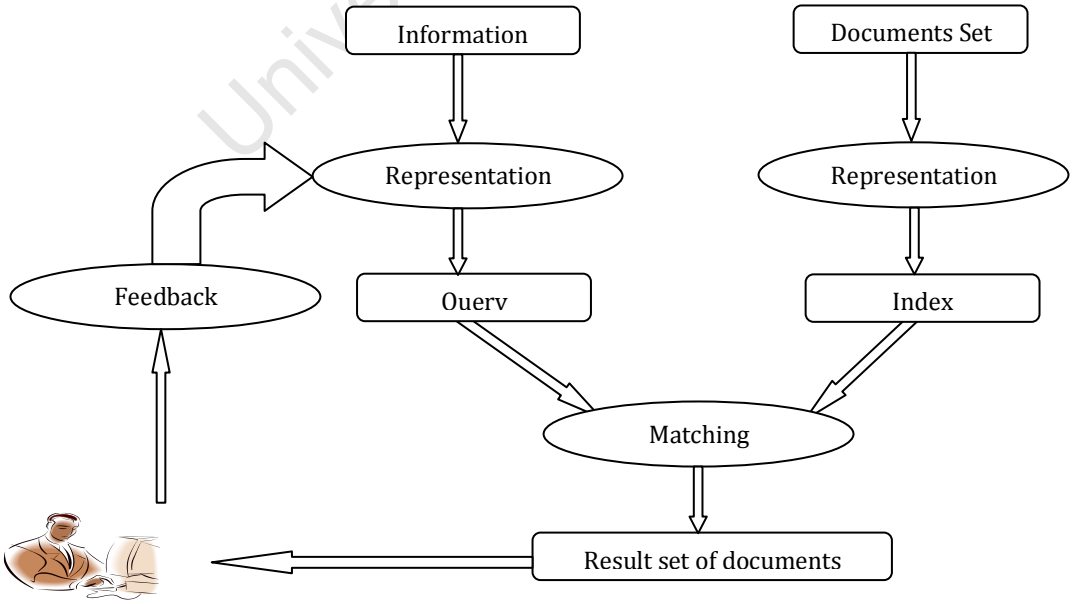


FIG. 2.1: A typical information retrieval task.

The broad process of IR, adapted from Nie (2010), is shown in Figure 2.1. The process of representing documents is called the *indexing* process (Manning, et al., 2008), in which keywords of documents are extracted. Such extracted keywords are known as *terms*. The term is the basic used unit for representing both documents and queries and it can be a word, phrase, stem, N-grams, etc, depending on what is needed from representing/indexing documents.

The process of producing index terms often goes through several operations, most of which are language-dependent. Examples of such processes are *tokenization* and *stemming*. Tokenization is the process of breaking a stream of characters into expressive and semantically meaningful pieces called *tokens*, whereas stemming renders different inflected and variant forms of a certain token to a single word stem(term). For instance, words like *participating*, *participates*, *participation* and *participant* may all be rendered to a common single stem *participat*.

The end product of the document representation process (indexing) is a new searchable structured description of documents in a form of a set of terms (index). A user information need is also represented in the same way so as to create a query, which searches against the created index. Thus, the matching process is usually carried out between a query (information need representation) and a set of represented documents.

In a broad sense and based on this matching, a set of documents with matched *scores* are often retrieved as a result for the matching process. A matched document score is used to determine the relevance matching degree of a document with regards to a query. The document is considered as relevant if it covers/addresses the user information need, rather than just containing query terms. Hence, a good IR system is expected to rank relevant documents on top and its quality is measured in terms of *precision* and *recall* . Precision can be though as the proportion of the returned documents that are relevant to the query, whereas recall is the proportion of the relevant documents in a collection that were retrieved by an IR system.

One might optionally need to re-formulate/feedback the original query so as to produce a better result list. In such a case the list of retrieved documents can be considered as an initial or intermediate result for the retrieval. This is known as *relevance feedback* (Rocchio, 1971). In relevance feedback the IR system may need the user to participate in the process (by judging which documents in the initial retrieved list are relevant) or the system could automatically perform this function based on top ranked documents. The latter approach is referred to as Pseudo Relevance Feedback (PRF). Regardless of the feedback approach, the IR system employs relevance feedback so as to re-formulate a new information need representation and then a second retrieval process using this new query is performed.

The above is a general overview of the retrieval process and there are still more details behind, many of which are covered in the following sections. However, the most important part of this process is what is called the *information retrieval model*. Retrieval models are described next.

2.1.2 Information Retrieval Models

A retrieval model (Manning, et al., 2008; Nie, 2010) is an abstract model/pattern for the information retrieval process whose function is to describe how an IR system represents documents and queries and how it predicts the relevance scores of retrieved documents with regards to a certain query during retrieval. Using retrieval models, the end product of a matching between a query and a set of documents is an ordered or unordered result set list, depending on the used retrieval model. Since retrieval models are abstract, most IR systems are built on the top of a certain retrieval model.

Retrieval models can be classified into two main types - these are the exact-matching and the best matching retrieval models (Belkin and Croft, 1992). The classification is derived from whether a document is exactly match the query (exact-matching model), and thus a document either matches the query or not, or a document matches the query to some relevance degree, and thus the IR model provides the best matching documents that match queries (best-matching model). The latter models are also referred to as the *ranked retrieval models*.

The exact-matching model usually utilizes a precise language, mostly with some Boolean operators, to build up the query and the result list is a set of documents with no ordering, as they exactly match the query. In the ranked retrieval model, queries are often free text words and the result set returns documents in a ranked list, mostly starting from the best matching document, as they are based upon frequency distributions of query terms in documents. The next sections discuss some of the common retrieval models in the two approaches.

2.1.2.1 Boolean Models

A Boolean model is an exact-matching retrieval model. In a Boolean model (Manning, et al., 2008), queries are formulated by a combination of their keywords (terms) with Boolean logic operators (AND, OR and NOT) in a precise language, in which these operators are handled during retrieval in a similar way to their use in conventional truth tables of the Boolean Logic. Likewise, documents are also represented as a conjunctive set of terms in a Boolean expression with a basic assumption that terms that do not occur in a certain document would not appear in its corresponding Boolean expression. Thus, a document is considered as exact-matched with regards to a certain query, if the terms that represent that document satisfy the Boolean expression representing the query and, hence, a set of matched documents can be obtained.

In conventional Boolean models, a result set is neither ranked nor makes use of frequency distribution statistics of terms in queries. But, some Boolean models allow non-Boolean operators like those used in proximity and wildcard operators, such as those used on current search engines. However, mostly document results are ordered chronologically, rather than an accurate estimation for their relevance degree to queries. In fact, ranking documents had been extended to Boolean models using different approaches. In such a case it is called the *Extended Boolean Model*. In extended Boolean models some

heuristics methods, like those utilizing fuzzy set theory (Paice, 1984) and those using some types of weights (Fox, 1983) are used.

The traditional Boolean model was used intensively by many commercial retrieval systems until 1990s (Manning, et al., 2008), as it can be effectively implemented, but yet it has some limitations. Firstly, the model does not provide ranked weighted documents. Instead, it centered around whether a document matches a query or it does not. This problem leads to higher precision, but low recall whenever the AND operator is used, and a low precision with a higher recall when the operator OR is incorporated in the query. Second, queries in Boolean models are relatively complex, especially when proximity operators are needed, and they cannot easily be constructed by normal users, unless complete knowledge about the collection, its index and its content is available.

2.1.2.2 Ranked Retrieval Models

Besides the complex formulation of queries, it was previously illustrated that conventional Boolean models do not rank documents according to their relevance level. This, forces users to explore all the retrieved documents or the most relevant documents are not found (Jackson and Moulinier, 2007). There is always a possibility to find such documents at lower ranks. Furthermore, the need of the Boolean models for expert users, at least in formulating queries, to build up the query in a very precise query language with operators diminishes their effectiveness for non-expert users (Jackson and Moulinier, 2007). This is because formulating queries in Boolean models can be a real burden for such users, who often prefer to type just free text queries consisting of just a few words and the retrieved documents should be ranked according to the relevance degree of these documents with respect to those users' queries. In such cases a ranked retrieval model is needed (Hiemstra, 2000) so as to estimate a relevance score for documents and to determine which of them is best matching the posted query. In that context, the ranked models have been shown to be more effective than Boolean models (Manning, et al., 2008). The next sections describe some of these ranked models.

2.1.2.2.1 Vector Space Model

The Vector Space Model (VSM) is one of the models that have a solid theoretical base (Salton, et al., 1975). The model derives its name from the fact that it represents both documents and queries as vectors in a common and high dimensional vector space. In that context, the set of documents often results in large vectors (for example, *n-value* vectors, where *n* is the number of distinct words in the collection under indexing). In this large vector space, each term in each vector is represented, in its simplest mean, in a binary form (1 for its presence and 0 for its absence in the document whose vector is presented in the vector space). However, instead of simply specifying 1 and 0 for the presence or absence of terms, terms frequencies (the total number of occurrences of terms in each document) can be considered. But, the major issue is what to use for weighting terms, as some terms can occur frequently while others do not. Accordingly, the most commonly method is based on the assumption that some terms may have

greater discriminating effect than others. Accordingly, most approaches utilize Term Frequency (TF) and Document Frequency (DF) computations. TF is the number of occurrences of a term in a certain document. This is a measure of aboutness. DF is the number of documents in which a term occurs and it is used for the purpose of determining term specificity on the basis that a term that tends to occur in many documents is less important than another term whose appearance in documents is infrequent.

The most standard weighting approach is the *TF. IDF* scheme, in which *IDF* (Inverse Document Frequency) is used to determine term importance as follows:

$$idf_t = log\left(\frac{N}{df_t}\right) \tag{2.1}$$

Where N is the total number of documents in the collection and df_t is the document frequency of the term t. In the standard weighting scheme, the term weight, denoted as w_t, in a document d_k is defined as a combination between its term frequency and its inverse document frequency, that is:

$$w_{t,d_k} = tf_{t,d_k} \times idf_t \tag{2.2}$$

Thus, terms in each document are assigned weights based on this standard approach. Terms in query are also assigned weights using the same standard method.

The closeness of document vectors to query vector (similarity matching) is often measured by using the angle size. Such angle can be computed as an inner product. Consider a text collection with m distinct terms t_j with $j = 1..m$. The extracted vector representation of a document d_k, denoted as a vector $\vec{d}_k$, would consist of m distinct terms (t_{1,d_k}, t_{2,d_k}, t_{3,d_k}, … , t_{m,d_k}). Note that in such perspective, the exact ordering of terms in a document is not considered. A similar argument applies to queries, that is a given query q can be represented as a binary vector $\vec{q}$ containing ($t_{1,q}$, $t_{2,q}$, … $t_{m,q}$). Since every term t_{j,d_k} in both $\vec{d}_k$ and $\vec{q}$ can be assigned a weight, denoted as wt_{j,d_k} and $wt_{j,q}$, respectively, then the inner product (similarity matching) between these two vectors ($\vec{d}_k . \vec{q}$) can be computed as:

$$sim(q,d_k) = \sum_{j=1}^{m} wt_{j,d_k} \, . wt_{j,q} \tag{2.3}$$

But, many terms do not occur in both the document and the query vectors, therefore, the similarity can be re-formulated as:

$$sim(q,d_k) = \sum_{j \in q} wt_{j,d_k} \, . wt_{j,q} \tag{2.4}$$

However, matching similarity in such a case may lead to significant problems. Since there may be a significant difference between a query vector and a document vector, due to short length of queries, long documents with a large number of terms may seem to be not relevance to many queries. Moreover, documents with similar vocabulary but with one of them having a longer length than others would probably have substantial difference in their vectors (Manning, et al., 2008). One proposed solution to these drawbacks is to apply a vector length normalization factor in order to normalize each vector to a

unit vector. The standard method used for vector length is the cosine similarity (Manning, et al., 2008). Assume that the angle between $\vec{d}_k$ and $\vec{q}$ is θ, then the similarity matching is applied as follows:

$$sim(q,d_k) = cos\,\theta = \frac{\vec{q}.\vec{d}_k}{|\vec{q}|.|\vec{d}_k|} \quad (2.5)$$

Where $|\vec{q}|$ and $|\vec{d}_k|$ are the Euclidean length of $\vec{q}$ and $\vec{d}_k$, respectively and $\vec{q}.\vec{d}_k$ is the inner product of the two vectors. Mathematically, the Euclidean length of a vector A (|A|) with n items, is defined as:

$$|A| = \sqrt{\sum_{k=1}^{n} A_k^2} \quad (2.6)$$

Accordingly, the similarity between the query $\vec{q}$, with l terms present, and $\vec{d}_k$, known also as the Retrieval Status Value (RSV), can be computed as:

$$sim(q,d_k) = \frac{\sum_{j=1}^{l} wt_{j,d_k}.wt_{j,q}}{\sqrt{\sum_{j=1}^{l} wt_{j,d_k}^2}.\sqrt{\sum_{j=1}^{l} wt_{j,q}^2}} \quad (2.7)$$

A number of alternative weighting schemes on *TF.IDF* were proposed in order to suppress the impact of skew weights. Such biased weights may result for different reasons. For example, longer documents may always have higher *tf*, and thus higher scores, as they contain more terms. Another example for biased weights may results from the fact that the significance of terms cannot always be expressed by just their number of occurrences, meaning that if the a particular term occurs m times in a document, this does not mean that all these occurrences are truly significant (Manning, et al., 2008). Several variants of standard weighting schemes were developed to improve on basic combination of the *TF-IDF*. For more details about these weightings refer to Salton and Buckley (1988).

2.1.2.2.2 Probabilistic Retrieval Model

The use of probability theory in documents retrieval originated with Maron and Kuhns (1960), who discussed that documents in a collection could be ranked according to their probabilities of relevance or their degree of similarity with regards to queries. The key issue here is how to compute a probability of each term in a query and how to assign the final probability that a document is relevant to the query. Broadly speaking, a probabilistic retrieval model employs the absence or the presence of a term in a document to predict a weight for that term. This weight corresponds to the estimated probability of relevance of that term and the combination of all the query terms' weights is thereby used to determine whether the document is relevant or not.

Given a query q and a document d_k in a collection consists of a set of terms $\{t_1, t_2, t_3, \ldots, t_m\}$, the set of terms in the document d_k can be represented as a binary vector $\vec{x}$: $(x_1, x_2, x_3, \ldots x_m)$, in which $x_i = 1$ represents the presence of term t_i in document d_k and $x_i = 0$ represents the absence of term t_i. In that

perspective, the probability that the document d_k, which is represented as a vector $\vec{x}$, is relevant to the query q, denoted as $Odd\ (R|\ q,\ \vec{x})$, in conventional probabilistic models (Robertson and Sparck-Jones, 1976), e.g., Binary Independence Retrieval (BIR) model, is computed according to document's odds of relevance, which is a ratio between the probability that the document belongs to relevant set of documents, denoted as $p(R|q,\vec{x})$, and the probability that the document belongs to the set of non-relevant documents, denoted as $p(\bar{R}|q,\vec{x})$. Formally,

$$Odd\ (R|\ q,\vec{x})\ =\ \frac{p(R|q,\ \vec{x})}{p(\bar{R}|q,\ \vec{x})} \tag{2.8}$$

Applying Bayes's theorem transformation, this ratio becomes:

$$Odd\ (R|\ q,\vec{x}) = \frac{p(\vec{x}|R,q)}{p(\vec{x}|\bar{R},q)} * \frac{p(R|q)}{p(\bar{R}|q)} \tag{2.9}$$

Where $p(R|q)$ and $p(\bar{R}|q)$ are the prior probability of retrieving a relevant document or non-relevant documents, respectively. Other symbols were defined above. Nevertheless, since $p(R|q)$ and $p(\bar{R}|q)$ are constants for all documents for a given query, and hence they do not affect the ranking, they are often eliminated from equation 2.9, resulting in:

$$Odd\ (R|\ q,\vec{d}_k) \propto \frac{p(\vec{x}|R,q)}{p(\vec{x}|\bar{R},q)} \tag{2.10}$$

But, based on independence assumption, e.g., the appearance or non-appearance of one term in a document has no effect on the presence or absence of any another term, the ratio of the probabilities of all terms in vector $\vec{x}$ occurring in relevant and non-relevant documents can be computed as the product of the corresponding ratio for each single term, thereby resulting in the following score, which is the matching score, denoted as $m\text{-}Score(d_k,q)$, between document and query:

$$m\text{-}Score(d_k,q) = \frac{p(\vec{x}|R,q)}{p(\vec{x}|\bar{R},q)} = \prod_{i=1}^{m} \frac{p(x_i|R,q)}{p(x_i|\bar{R},q)} \tag{2.11}$$

Since x_i is either 0 or 1, terms can then be separated. Such separation yields (Robertson and Spärck-Jones, 1976):

$$\frac{p(\vec{x}|R,q)}{p(\vec{x}|\bar{R},q)} = \prod_{x_i=1} \frac{p(x_i=1|R,q)}{p(x_i=1|\bar{R},q)} * \prod_{x_i=0} \frac{p(x_i=0|R,q)}{p(x_i=0|\bar{R},q)} \tag{2.12}$$

Where:

- $p(x_i = 1|R,q)$ is the probability of a term occurring in a document relevant to query.
- $p(x_i = 1|\bar{R},q)$ is the probability of a term occurring in a document non-relevant to the query.
- $p(x_i = 0|R,q)$ is the probability of a term does not appearing in a document relevant to query.
- $p(x_i = 0|\bar{R},q)$ is the probability of a term being absent in a non-relevant document to query.

But, for simplifying the formula assume that $p_a = p(x_i = 1|R, q)$ and $u_a = p(x_i = 1|\bar{R}, q)$. With the assumption that terms that do not appear in the query (q_i=0) would result in $p_a = u_a$, only terms that are present in the query will be considered in the product of odds in equation 2.12. Such simplification would result in the following formula (Manning, et al., 2008):

$$\frac{p(\vec{x}|R,q)}{p(\vec{x}|\bar{R},q)} = \prod_{a:x_i=1,q_i=1} \frac{p_a}{u_a} * \prod_{a:x_i=0,q_i=1} \frac{1-p_a}{1-u_a} \quad (2.13)$$

The first product in this formula is over query terms which appear in the document, whereas the second product is over query terms which do not appear in the document. But if the right product is to be over all query terms, then equation 2.13 would result in the following:

$$\frac{p(\vec{x}|R,q)}{p(\vec{x}|\bar{R},q)} = \prod_{a:x_i=1,q_i=1} \frac{p_a(1-u_a)}{u_a(1-p_a)} * \prod_{a:\, q_i=1} \frac{1-p_a}{1-u_a} \quad (2.14)$$

In this formula, the right product is over all query terms. Accordingly, it will be constant for a given query. This results in the need to only estimate the right product. The resulting quantity after taking the log of this product is thus, used in ranking documents and computing RSV:

$$RSV(d_k,q) = \sum_{t_a \in q} \log \frac{p_a\,(1-u_a)}{u_a\,(1-p_a)} \quad (2.15)$$

In the equation, $(p_a/1-p_a)$ is the odds of query term occurring if the document is relevant, whereas $(u_a/1-u_a)$ is the odds of query term occurring if the document is non-relevant. However, to estimate these probabilities, sample documents that are judged are to be employed. In such a case document statistics are used. Assume that the total number of documents in a collection (sample judged) is *N, R* is number of relevant documents in sample set, *r* is the number of relevant documents containing the query term and *n* is number of documents containing the term. Using these assumptions, the following weightings can be used to estimate p_a and u_a in equation 2.15 (Robertson and Spärck-Jones, 1976):

$$wt_1 = log\left(\frac{\frac{r}{R}}{\frac{n}{N}}\right) \quad (2.16)$$

$$wt_2 = log\left(\frac{\frac{r}{R}}{\frac{n-r}{N-R}}\right) \quad (2.17)$$

$$wt_3 = log\left(\frac{\frac{r}{R-r}}{\frac{n}{N-n}}\right) \quad (2.18)$$

$$wt_4 = log\left(\frac{\frac{r}{R-r}}{\frac{n-r}{(N-n)-(R-r)}}\right) \quad (2.19)$$

Thus, term weights in classical probabilistic models can be computed. However, since a zero possibility may occur, e.g. when a specific term does not appear in all relevant documents, a constant value of $^1/_2$

is added as a form of smoothing (Robertson and Spärck-Jones, 1976). Such smoothing results in weighting functions, for wt_4 for example, as:

$$wt = log\left(\frac{\frac{r+0.5}{(R-r)+0.5}}{\frac{(n-r)+0.5}{(N-n)-(R-r)+0.5}}\right) \quad (2.20)$$

However, in reality relevant and irrelevant documents, e.g. the value of R, will not be available priori. Accordingly, some assumptions can be made (Manning, et al., 2008, Nie, 2010). For example, it can be assumed that number of relevant documents is very small compared to irrelevant documents. Using such an assumption the probability would be computed from only non-relevant set of documents. Accordingly, the probability of a term occurring in non-relevant documents, $p(x_i = 1|\bar{R}, q)$ or u_a, $log(1- u_a)/ u_a$ = log (N-DF)/DF) $\approx log\,(N/DF)$, where DF is the document frequency of term.

Term frequency and document length did not appear in the classical probabilistic models. In particular, they only take into account the document frequency component in documents with consistent length. In addition, they are based on simple presence and absence of a query term in documents. Considering the magnitude of both term frequency and length of documents in computing scores, the Best Matching weighting schemes (BM), which are based on a 2-poisson distribution, were developed (Robertson and Walker, 1994) as an extension for probabilistic model. Among these schemes, the BM25, also known as the OKAPI, is the most widely-accepted model by the information retrieval community (Manning, et. al, 2008). The model considers both term frequencies in documents and collection statistics. It also attempts to manage length of documents using some averaging technique. In particular, in BM25 larger values of occurrences of terms (TF) do not increase weights at the same rate of what smaller values do as weights of the latter values are approximately linear (Hiemstra, 2000).

Accordingly, in BM25 the term weight $wt_{t,D}$, in a document D, which incorporates both document length normalization and term frequency component, is computed as follows:

$$wt_{t,D} = [\frac{(k_1+1)tf_{tD}}{k_1((1-b)+b\,\frac{L_D}{avg\,L_D})+tf_{tD}}] \quad (2.21)$$

Where

- tf_{tD} is the frequency of term t in the document D.
- L_D is the length of document D.
- $avgL_D$ is the average document length across the collection.
- k_1 is a parameter used to tune term frequency in a way that large values tend to make use of raw term frequency. For example, assigning a zero value to k_1 corresponds to not considering the term frequency component, whereas large values correspond to raw term frequency. k_1 is usually assigned the value 1.2.
- b is another free parameter where $b \in [0,1]$. The value 1 means to completely normalizing the term weight by the document length. b is usually assigned the value 0.75.

Using this formula (2.21), the term frequency component increases modestly as the value of the frequency increases because long documents contain usually multiple appearances of terms. Thus, the use of the average length of documents would likely credit shorter documents.

With respect to relevance weight (document frequency estimation), it is possible that the relevance information may be unavailable as illustrated above. In such a case *R=r=0*. Accordingly, wt_4 (Robertson and Spärck Jones, 1976) can be reformulated as:

$$wt = IDF = log\frac{N-n+0.5}{n+0.5} \tag{2.22}$$

In which both *N* and *n* are defined above. Thus, in BM25, the *RSV* of document *D* against the query *q*, denoted as $wt_{q,d}$, is computed as:

$$wt_{q,D} = \sum_{t\in q}\left[log\frac{N-n+0.5}{n+0.5}\right].\left[\frac{(k_1+1)tf_{tD}}{k_1((1-b)+b\,\frac{L_D}{avg\,L_D})+tf_{tD}}\right] \tag{2.23}$$

A similar term weighting can be also used when queries are long. In such case, formula 2.23 becomes (Spärck Jones et al., 2000):

$$wt_{q,D} = \sum_{t\in q}\left[log\frac{N-n+0.5}{n+0.5}\right].\left[\frac{(k_1+1)tf_{tD}}{k_1((1-b)+b\,\frac{L_D}{avg\,L_D})+tf_{tD}}\right].\left[\frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}}\right] \tag{2.24}$$

In which tf_{tq} is the term frequency of the term *t* in the document *D*, k_3 is another parameter to tune term frequency in query *q* and other symbols are as defined above.

Extension of the BM25 to Multiple Weighted Fields

Documents are often found in structured forms. For instance, text in documents may be subdivided into several fields, e.g., title, body. Thus, a question may rise: how to apply a ranking function model designed for unstructured documents, as in the BM 25, to structured documents (Robertson, et al., 2004). In other words, how should scores of fields in structured documents be combined together so as to compute final scores of documents - bearing in mind the fact that terms of queries may match some/all fields.

The most common approach is to compute a score for each textual field as if that field type is an isolated collection of unstructured documents (Robertson et al., 2004). Next, a linear combination of these scores is performed. Consider a set of structured documents, each of which is denoted by *D* with *n* fields and a standard weighting function *wt*. A typical linear combination of fields in structured documents can be defined as:

$$wt_{q,D} = \sum_{f=1}^{n} wt_{q,f}\ .\ v_f \tag{2.25}$$

Here, $wt_{q,f}$ is the computed weight of the terms of the query q in a particular separated field f in the document D, v_f, is the corresponding field weight (i.e. it might be desired to weight the title field with 3 and the body field with 1) and $wt_{q,D}$ is the linear combination of the fields' scores, which are obtained from scoring each field separately and this would result in the final score of the structured document D.

But, using such a linear combination has been criticized by Robertson, et al. (2004), who proposed an extended version of the BM25 weighting scheme in terms of multiple weighted fields. For example, one problem in linear combination is the issue of choosing collection statistics, e.g., document frequency of terms in linear combination is computed from a specific field only and, hence, if the field is short then statistics may be quite unstable. Similar drawbacks in linear combination of fields' scores occur also in term frequency and document length components and in tuning parameters, as well, which are to be set empirically for every field in the structured documents and, hence, a large number of tuning parameters is to be set.

To suppress the effects of such difficulties in linear combination of field scores, the extension of the BM25 in Robertson's and his colleagues work is based on refraining from doing linear combination of scores obtained from scoring every field in documents. Instead, the proposed alternative is to calculate a single score for the linear combination of term frequencies of terms in the different fields, but weighted by the corresponding weighted fields. The scoring function in this way is applied only once.

This means that from the original structured document a new non-structured one with new term frequencies, which result from the combination of the original term frequencies in the different fields, is obtained. The linear combination is, however, weighted by the fields' weights. For example, assume that the abstract field is desired to be weighted by 3, whereas the weight 1 is assigned to the body field. Using the developed approach, such a document would be replaced by the same document but with the abstract repeated 3 times and a single score for term frequencies is computed. The same arguments were also applied to the rest of the components in the ranking function. For example, in the new produced document, document frequency of a particular term would be the document frequency of that term if different fields are merged together.

2.2 Cross-Language Information Retrieval: Current Approaches

In monolingual IR, both queries and document collection are represented in a single language. But, it might happen that a user may need to post his query in a language different from the document language, probably because he may not be sufficiently fluent to construct meaningful and reasonable queries in the document language (Gey, et al., 2005). For example a user may submit the query in the Arabic language, whereas the result set is obtained in English. Such a special case of IR is known as Cross-Language Information Retrieval (CLIR). The primary goal of CLIR is to allow users to post queries in one language (known as *source language*) and retrieve documents in another language different from the query language (known as *target language*). The implicit assumption made here is that users do understand results obtained in another foreign language (i.e. English) or they do understand more than

one language but, they are not able to express their queries in the documents' language and, thus, they prefer to write their queries in their native languages.

The underlying assumption behind CLIR makes crossing the language barrier a fundamental and more challenging task than the monolingual IR as additional difficulties are approached beside those inherent in traditional monolingual searching. The basic difficulty, however, is that unless a similar representation for both the query and documents set is utilized, direct matching between their representations usually fails, as they are described into different languages. Thus, in order to satisfy this requirement, the majority of the CLIR systems adopt some types of translation to translate the query, the document or both before the matching between them takes place. Accordingly, the major component in CLIR system is the translation process (and its consequences such as the weighting problem).

According to Grossman and Frieder (2004), the major critical aspects in CLIR that are related to translation module were addressed by Oard, who raised three interrelated questions:

1. The first question is what to translate. This is a question about either to translate the query into document language, the document set into the query language or both of the query and the document set into a single language.
2. The second question is what types of terms (i.e. stemming, words, n-gram(s), etc) should be translated. This is related to some types of pre-processing and processing approaches in different languages. Since a translation process should be performed in most approaches to CLIR, the form of the translated word could be significant.
3. Finally the third question is how some obtained candidate translations for a single source term, using some translation source, can be utilized in the retrieval process. This is the term weighting problem in CLIR, in which it is important to weight the alternative target translations of a single source term in order to suppress the effect of bad or noisy translations. For instance, it might be reasonable to weight some translations higher than others.

It is obvious that the first two questions are interrelated to another question, which will be considered as the fourth one in this thesis, that is which translation source, and approach as well, can be used/acquired so as to perform translation process. Numerous translation resources are available and/or needed. However, the availability of translation resources is often dependent on the targeted language. For instance, sufficient translation resources are available for English in many domain specific fields and/or general domains, while Arabic shows lacking in such types of resources (Alansary, et al., 2007, 2008).

The next sections explore the research findings in the IR community to answer the above mentioned questions, starting with whether to translate query, documents or both.

2.2.1 Query Translation versus Document Translation

To accomplish the task of matching between queries and documents in CLIR, a typical strategy is to either translate documents, queries or both, as was mentioned above. The role of performing such translation, whether it is done over documents or queries, will result in shrinking the CLIR process to a monolingual IR and thus, a straight matching between queries and documents can be carried out.

However, each of the translation approaches (the document versus the query) has some cons and pros (Zhang and Lin, 2007). On one hand, the query translation approach is relatively easy, fast and it does not require much effort. Furthermore, it is less expensive in terms of computational cost, when it is compared with document translation, because it doesn't require any modification for the created index of document collection. Thus, if the index grows no modification on the index will occur, unlike in document translation approach, which requires the index to update with each new translated document. Nevertheless, translating the query for the CLIR task often has a major weakness, that is a query is usually short and precise with limited context. This usually leads to some kind of ambiguity, which occurs as a result of the limited context and the several produced candidate senses, which in turn makes the confidence about translation quality much lower. Therefore, many studies focus on this translation ambiguity difficulty, as it will be seen later in this chapter.

On the other hand, translation of documents into the query language appears to be more efficient because documents usually provide full and rich context, and hence they increase the certainty about translation quality and the translation ambiguity might be reduced to its lowest levels. Other benefit for the document translation in CLIR is that in most cases the translation workload is transferred to indexing time, instead of real-time (during ad-hoc retrieval) as it happens in the query translation approach. Nevertheless, document translation approach is a significantly time-consuming process that usually incurs much computational cost (Nie, 2010). This difficulty may prevent exploring much of the available context in documents, especially when a large number of documents are needed to be explored and/or translated. Another additional drawback for the document translation approach is that the target language(s) of each document in the document collection should be specified early in order to translate all the documents in these target languages before the index being created (Nie, 2010). This is a crucial problem. First, such an approach is impractical in real multilingual environments because it usually requires each document to be translated in all the desired languages and consequently massive storage space is required. Second, it is unreasonable to restrict users to a limited number of languages, which were specified early, in particular to those languages whose documents were translated and stored since in CLIR retrieval users might submit a query in any language, regardless of document collection language. Furthermore, it is not viable, impractical and computationally expensive in ad-hoc retrieval to translate large corpora. Accordingly, Chen and Gey (2004b) implemented a fast version and an approximate method for translating documents, so as to overcome this problem. In results, they found that fast translation of documents using some kind of bilingual wordlists produced by machine translation systems (see section 2.2.3.2) is as efficient as the query translation, while the combination between them (fast translation of documents and query translation) is significantly better than each of the two approaches when each of the approaches was implemented separately.

McCarley (1999) exploited also the use of a hybrid approach that merges both document translations, from one direction, with query translation, from the other direction. This is applied by computing final scores of documents as an arithmetic mean between the mutual result scores that are obtained from each unary translation direction. In experiments, the study used the English-French TREC 6 and TREC 7 collections. Results reported in the experiments showed that applying such a merging technique obtained

statistically significant results, in terms of average precision, compared to the translation in one direction, especially the query translation based approach, even if the translation of the query is robust. Similar results were also concluded by Chen and Gey (2004a) and Aljlayl et al. (2002). Nevertheless, McCarley (1999) study indicated also that the benefit of merging both document translations with query translation is totally reliant on the translation direction between languages, meaning which language is being used in the query and which is in the document and whether it is needed to make document translation or query translation. This is obvious from the results, which showed that French-English translation is much better in performance than English-French translation, whether it is used in query translation or document translation. The study justified this phenomenon because of the structure of the languages, e.g. the use of phrases in French.

In spite of the beneficial effects of using the document translation approach, separately or with query translation, it is impractical to perform a full document translation for large collections, such as on the Web. Furthermore, translating queries in a language that is not determined in advance is cheaper, elegant and simpler, as inexpensive translation resources like dictionaries are often utilized. Therefore, it is concluded in the IR community that query translation is the most viable and flexible option and, thus, it is the most dominant approach in CLIR (Kishida, 2005; Nie, 2010), rather than document translation.

2.2.2 Text Processing in CLIR

As in monolingual IR, various processing strategies in CLIR are utilized for extracting index terms in documents (document representation) or queries. Although, these strategies were previously known in monolingual IR, they are common in CLIR for two reasons. First, Most of the IR research techniques were originally introduced for the English language, with English and its peer European languages in mind (Kishida, 2005, Nie, 2010). With the advent of the WWW, however, and the appearance of a wide variety of resources, a growing need for CLIR utilities for non-European languages, such as the Arabic and Asian languages, have shown a new challenge for the approaches of processing texts, as this task is specifically a language-dependent function. For example, some languages do not make use of space delimiters, while others do. Second, the effect of text processing has a significant impact on the translation process. For instance, a particular translation resource may fail to provide senses for a certain source word due to ineffective stemming, and thus matching, although some/all of these word's senses may be listed in its entries. For these two reasons, which were emerged with CLIR, processing texts become a considerable part of the CLIR process. In that context, processing text in CLIR answers the question of what type of tokens can be utilized to perform translation, which was the second question in translation-related aspects. Among the several processing strategies, tokenization, stopwords removal and stemming are discussed below.

2.2.2.1 Tokenization

Tokenization is the process of breaking a stream of characters into expressive and semantically meaningful units called tokens and possibly omitting some particular characters - such as punctuation. Tokenization performs numerous tasks. Examples include, but are not limited to, identifying acronyms, punctuation marks – such as periods and hyphens, manipulating words to lower case and retrieval of words that are segmented across lines. In addition to these tasks, tokenization may detail some positional information about word occurrences in documents. The straightforward output of the tokenization phase is that words in documents are segmented and identified from one another and be ready for indexing, as in the IR process.

At the first glance, tokenizing text appears trivial since tokens, as in many languages such as the Arabic, are isolated by whitespaces. However, such a simple approach may be error-prone (Manning and Schütze, 1999; Manning et al., 2008). Since a token can be a word, an acronym, a punctuation mark, hyphenation-separated words, etc, different possible segmentations can take place and thus invalid segment may occur. For example, what scenario that would take place if the tokens 'Ali's home' or 'language-dependent' appears? In such cases one might consider the 'language-dependent' words as one token, while another may use the hyphen as a delimiter for the word end. Another example is the case of joining the Arabic preposition ك with the word ريم (meaning: antelope), resulting in the word كريم. In this word, there is no explicit boundary between the word ريم and the preposition ك. Therefore, the word كريم can have two different meanings: if it is used as a single word, it means generous; and if it is used as a sentence, combining the preposition ك and the word ريم, it means 'as an antelope'. Thus, combination of prepositions and words is a challenge for Arabic tokenizers and may add ambiguity to information retrieval.

Due to such types of difficulties, tokenizers are language dependent tools. In a language such as the Arabic, the white space delimiter may be sufficient. However, this linguistic property does not hold for many languages. For example, in most East-Asian languages, such as Japanese, there is no clear boundary or white space between words. In such cases an early process of word segmentation is required. But, a certain word can be split at different positions. Therefore, numerous approaches have been developed for segmenting words in East-Asian languages. A typical technique for such a decompounding process, which is basically a Natural Language Processing (NLP) problem, makes use of dictionaries that contain all the possible segments, (Nie, 2010). In such a typical technique, the words are split firstly into the different possible ways and, next, a left-to-right matching process for each segment with the dictionary is carried out and the longest matching in the dictionary is chosen. However, such methods are error-prone because many segments may not be covered in the dictionary. Moreover, there may be several possible segments for a given sentence. From a CLIR perspective, such mistakes might degrade the translation effectiveness, which in turn degrades CLIR performance. Other approaches (i.e. Chen and Gey, 2004b) also estimated probabilities for each possible segmented word when word usage statistics are available. Then, the sequence of segments which result in the highest likelihood would be selected.

However, *compound words* may result in wrong tokenization. Compound words in many languages are usually composed by joining two or more words together directly, as in the word 'birthday', or using a hyphen, as in the case of 'language-dependent' (Pirkola et. al, 2001). This linguistic feature occurs frequently in some languages such as Finnish and German (Pirkola et. al, 2001). Therefore, for such languages the tokenization process becomes more complicated because a decompounding process is needed so as to split a certain compound into its constituents. However, the problem here is how to choose the most probable constituent word among all the possible constituents in a compound. For instance, some approaches (Chen and Gey, 2004b) attempt to produce firstly all the possible ways to decompose a certain compound in German using a base dictionary. Next, the decomposition with the smallest number of components, if any, is chosen. If there is more than one decomposition with the same number of smallest components, a probability for each decomposition is computed using the product of the relative frequencies of the constituents' components in a collection in the language whose text is under processing. Hence, the most probable constituents in a compound are those whose likelihood is maximized. Apparently, since decompounding is somewhat similar to the segmentation of words, somewhat similar approaches, too, are employed for the former decompounding.

2.2.2.2 Stopwords

Stopwords are words with little value when conducting a search as they would be occur very frequently in documents. They are typically prepositions (i.e. *by*), articles (i.e. *an*), pronouns (i.e. *it*), etc. From an IR prospective, such words are usually not being indexed. It was previously shown that words that are most frequent (those with high document frequencies) cannot distinguish between documents and, thus, they are deemed to be of little importance. Stopwords have the same feature, and thus, they are usually eliminated. However, as reported in Savoy (2007), Moulinier illustrated that some IR systems assume that it is always deemed important to index all words, including stopwords, in documents and they only eliminate stopwords in queries. This is to avoid causing the system to erroneously eliminate keywords like *vitamin a* or *US navy*, in which both the article *a* and the pronoun *us*, respectively, would be removed if a stopword list is used.

It was also shown that it is not always correct to remove stopwords early (Al-shammari and Lin, 2008a, 2008b), especially in the highly inflectional languages such as Arabic. Stopwords in such languages are often helpful in determining POS in words that immediately follow the stopword and, thus, they may help a lot in the selection of the appropriate stemmer. For example, the syntactic categorization of the subsequent word after the Arabic stopword بعد (meaning: after) is the noun and, thus, such a noun can be lightly stemmed, unlike verbs which can be either lightly or heavily stemmed. Regardless of the used approach, from the CLIR view, stopwords are often removed before translation. More details are provided for Arabic stopwords in the next chapter.

2.2.2.3 Normalization and Stemming

After a text has been broken up into tokens (tokenized, even if segmented), those tokens are usually normalized. Normalization is the process of producing the canonical form of a token in order to maximize matching between a query token and document collection tokens. In its simple form normalization pre-processes tokens to a single form, but very lightly. This is often done in several pre-processing stages. One common approach in normalization, for example, is to remove a certain character/symbol from a token, e.g., removal of the hyphen or any non-character symbol. Another approach converts all words in a text written in Romanized forms to a single case (known as *case folding*) - lower case for example. But, the normalization is also a language-dependent process. For example, in the Arabic language, normalization is used to render different forms of a particular letter to a single Unicode representation, e.g., replacing the Arabic letter un-dotted ى with a final dotted ي, when this letter appears at the end of an Arabic word. Normalization in Arabic is discussed in next chapter.

Normalization is also employed to morphological analysis, e.g. for recognizing POS of words. In such a case normalization is performed without collapsing POS variants of words. However, in its brutal forms, normalization is used to handle morphological variation and inflation of words (Levow et al., 2005). This is called *stemming*. Since documents and/or queries may have several forms of a particular word, stemming is the process of mapping and transforming all the inflected forms of a word into a common shared form and, thereby, this shared form would be the most appropriate form for indexing the representations and for searching as well. For example, using stemming for English documents, IR systems are able to retrieve all documents that contain inflected words like *play*, *plays*, *player* and *playing*. In monolingual IR, stemming appears to have a positive impact on recall more than precision (Kraaij and Pohlmann, 1996; Pirkola et al., 2001) because a large number of relevant documents would likely be retrieved, although they may not be ranked at the top of the retrieved results. Furthermore, stemming shows a high positive effect on highly inflected languages, such as Arabic (Pirkola et al., 2001). An additional advantage for the stemming is that it also reduces the size of the index since many words are grouped together in a single canonical form.

A number of studies have been devoted to stemming for a wide range of languages and different approaches were proposed. Examples (Larkey et al., 2007) include light stemming, statistical-based stemming using N-grams or parallel corpora (collections), morphological analysis and co-occurrence analysis. Some of these stemming approaches are language-dependent (i.e. morphological analysis) while others, such as the statistical-based methods and co-occurrence techniques, provide more language independency. However, in spite of the large number of techniques for stemming, two major approaches are the most dominant. These are light stemming (known also as affix removal stemming) and heavy stemming (morphological analysis stemming). The light stemming chops off some affixes – such as plural endings in English - lightly from words and without performing deep linguistic analysis, whereas the second technique, which is heavy stemming, known also as the root-based approach, performs heuristic and linguistic processes so as to extract the root of a word. Each of the two types has some pros and cons. On one hand, heavy stemming retrieves all the related text and reduces the index size significantly.

Furthermore, the approach also maintains Part-Of-Speech (POS) distinctions (Levow et al., 2005), but it may erroneously cluster some different words into a single root, known as the over-stemming problem, leading to a low precision. On the other hand, light stemming achieves the goal of retrieving the most pertinent documents, but it may not succeed to cluster semantically similar words together, known as the under-stemming problem, resulting in low recall. But, since light stemming may fail to preserve parts of speech (Levow et al., 2005), e.g., there may be a noun and a verb that are both stemmed to a single stem, it increases the possibility of matching between stemmed documents and stemmed queries. Additionally, Paice (1994) had shown that light stemming moderates the over-stemming errors but it may result increasing the under-stemming errors, while heavy stemming reduces the under-stemming errors while it may result in increasing the over-stemming errors.

Motivated by the above drawbacks of both light and heavy stemmers and the fact they are language-dependent, statistical-based stemmers that demonstrate as language-independent techniques to conflation were also proposed. Examples of statistical stemmers are those based on corpus analysis (Xu and Croft 1998; Larkey, et al., 2002). The basic principle behind the statistical corpus-based stemmers is that since conflated words in a given corpus in a certain domain tend to co-occur with words in the same corpus in that domain, then the relationship between words can be utilized to prevent, for example, two semantically different words with the same stem being grouped together. Based on this argument, Xu and Croft proposed a statistical stemmer that makes use of the co-occurrence statistics extracted from a corpus-based analysis in order to create associations among words of the same domain. In particular, Xu and his colleague used a heavy stemmer to produce sets of classes, at first, with each class containing a number of words that were grouped into a single root, even if they are over-stemmed (i.e. police and policy would be in the same stem class). Following this, the technique reduces the several equivalence classes, by computing co-occurrences between word pairs in the same initial class. Hence, the equivalent classes are re-grouped according to these co-occurrence relationships. Using both English and Spanish, results of this work showed that the approach is effective for improving stemming.

Statistical stemmers based on exploring the structure of words in raw texts, and in a way similar to corpus-based analysis, were also investigated in CLIR (Buckley, et al., 1995). For example, a statistical stemming technique based on frequencies of words and a statistical rule induction was proposed to automatically identify information on common candidate suffixes from a text collection (Oard, et al., 2000). Hence, the most frequent suffixes are utilized for the purpose of stemming the input words. Applying such statistical stemmers to some European languages with backoff translation, which is discussed next in this section, the results in Oard's work showed that the approach yielded significant effectiveness over an un-stemmed approach.

Examples of stemmers are the Light10 (Larkey, et al., 2007), the Porter stemmer (Porter, 1980) and the Krovetz stemmer (Krovetz, 1993). The former is an Arabic light stemmer that is widely used for Arabic documents while the latter ones are rule-based English stemmers. In the Snowball project[4], a number of stemmers for European languages were developed and made available.

[4] http://www.snowball.tartarus.org/

From a CLIR perspective, in which queries and/or documents are to be translated, stemming can be done into two different approaches: before translating the query and/or document or after translating them. Post-translation stemming seeks to match words in documents with those in a query, as both are expected to have a common shared stem for the several inflected forms of each single word and, thus, stemming could balance for the vocabulary mismatching between documents and queries. The situation here is similar to stemming in monolingual IR. Pre-translation stemming helps in maximizing coverage of a translation. In particular, if the direct mapping between an inflected source word under translation and words in a particular translation resource does not match, their stemmed forms may match. In fact, in such matching of the words in a translation source and word to be translated, stemming may be useful when it is implemented steadily. This would enhance the coverage of the translation resource. Such gradual implementation of stemming is known as the backoff translation technique (Oard, et. al, 2000). In the backoff translation technique four possible succession combinations exist so as to moderate the problem that an exact translation is not covered in a translation resource. First, a matching process between the surface form (un-stemmed) of the query/document with the surface forms of the translation resource is performed. If this procedure does not succeed, the surface form of the query/document is stemmed in order to match this stem with the surface forms of the translation source. If this also fails, then the surface form of the query/document may match with the stemmed forms of the resource. If this still does not work, then stem both the surface form of the query/document and the surface forms in the translation source and perform a matching between these stems. Thus, using such a successive technique of backoff translation seeks to provide selective translations and hence ambiguity is minimized. Furthermore, the technique has shown to be efficient for retrieval effectiveness and was used in the CLEF (Cross-Language Evaluation Forum).

The type of stemmer also may affect the CLIR process. For instance, whenever a bilingual dictionary is employed for translation, a root-based stemmer probably results in producing too many alternative senses/translations. Most of them may be invalid or superfluous translations, especially in those highly inflected languages, such as Arabic (Larkey et al., 2007). Thus, when the result is retrieved using these translations, many documents would probably be irrelevant.

2.2.3 Translation Resources

Numerous bilingual/multilingual resources may needed to be acquired for CLIR systems so as to list or identify sense(s)/translation(s) for a source query term. Some approaches, like dictionaries, explicitly list the entries, while other resources can list senses implicitly, e.g., in *corpora*. A corpus is a repository of collected natural language materials such as textual information, which contains different types of information in different formats (Manning, et al., 2008).

This section is devoted to answering the fourth question, which was which translation source, and approach as well, can be used/acquired in order to successfully perform the translation process. Basically, four major resources are often utilized for translation. These are: machine readable

dictionaries, machine translation tools, parallel or comparable corpora and the utilization of the Web for extracting translations.

2.2.3.1 Dictionary-based Approach

The dictionary-based approach is the most widely used technique for the CLIR process (Nie, 2010; Ture, et al., 2012). It has been much explored in the CLIR community and has been used in several CLIR experiments (Ballesteros and Croft, 1998; Pirkola, et al., 2001; Levow, et al., 2005; Zhou et al., 2008). As dictionaries often list a large number of entries sorted alphabetically in bilingual or multilingual forms, the dictionary-based approach for translation replaces each term or phrase, mostly in queries, in a source language with its corresponding translation candidate(s) in a target language, using a Machine Readable Dictionary (MRD), which is an electronic copy of its corresponding printed dictionary. The most dominant approach when the MRD is used is the word-by-word translation (Ture et al., 2012). This makes it easy to substitute each term in a query with its corresponding translations, which may be placed into a bag of words in the target language, for example.

As general dictionaries may result in several senses, especially if the term to translate is technical, and/or may fail to provide senses, translation ambiguity may occur. Therefore, some approaches advocated the use of some specialized dictionaries, beside those that are general, (Pirkola et al., 2001). This is because domain-specific dictionaries provide usually 1-2 translations, depending on language and domain, for a given word in a source language and, thus, they reduce ambiguity during translation. In contrast, general dictionaries may have several translations enumerated under a given source word. Furthermore, specialized terms that are obtained from domain-specific dictionaries are usually good candidates to be search keys.

Two approaches (Pirkola, 1998; Pirkola et al., 2001) can be used when both specialized and general dictionaries are employed for translations: sequential translation and parallel translation. In the former approach, as the name indicates, a translation process that is based on a general dictionary is applied only when a given source term is untranslatable from a domain-specific dictionary, whereas the latter approach translates terms in parallel from the two dictionaries and, hence, all terms will be considered in the final target query. Both parallel and sequential translation approaches were applied by Pirkola (1998) using Finnish to English CLIR with medical MRD and general MRD for Finnish - English. The study showed a significant impact on retrieval and the special dictionaries have a positive effect on MRD translation but the effectiveness depends on the text domain that is being searched with the translated queries (i.e. medical, technology, news) and the method in which the special and general dictionaries are utilized. The importance of Pirkola's study is that it shows in most cases that specialized dictionaries provide valid and unambiguous translations for scientific terms, which are the most important search keys. Furthermore, the combination of both specialized and general dictionaries can be applicable to any type of specialized queries.

The relationship between sizes of dictionaries in terms of their entries and the impact of using different sizes on effectiveness of CLIR systems was also explored. In empirical English-Arabic CLIR and English-

Chinese CLIR experiments, Xu and Weischedel (2005) reported that the improvement of CLIR increases as the size of dictionary increases. But, after reaching 10,000 entries in dictionaries, the retrieval performance stays at the same level. This is especially true when including the translations for the 10,000 most frequent English terms. Hence, extending entries of the dictionary to contain more words does not mean that the effectiveness of the CLIR system would increase. Nevertheless, as stated by Nie (2010), it is expected that increasing the number of words in dictionaries would probably have a positive effect on CLIR effectiveness.

However, dictionary-based approaches for query translation also have some major defects (Ballestros and Croft, 1997; Ballestros and Croft, 1998; Pirkola et al., 2001; Nie, 2010) that may degrade their performance. One such drawback is that a translation may be awkward. This is because entries may not match the source terms unless both those source terms and the entries are normalized or stemmed. Furthermore, it is common in a dictionary-based approach that there are many extraneous and superfluous translations for a single source term, resulting in translation ambiguity. In fact, translation ambiguity may result for different reasons (Pirkola et al., 2001). Examples include *homonymy* (a source word with more than one unique meaning), *polysemy* (a single word can have more than one distinct meanings, but these meanings are related, e.g., the *head* of the department and the *head* of the body); and the absence of phrases. Approaches to translation disambiguation are explained in section 2.2.4.2. In addition to these drawbacks, which are not limited to dictionary-based approaches only, dictionaries may be limited in their coverage and many words may not be listed in their entries, especially for languages with few linguistic resources, and hence resulting in what is known as the *OOV* (Out-Of-Vocabulary) terms. The OOV problem and its resolution are discussed in section 2.2.4.1.

In spite of these problems, which were mitigated, as it will be discussed later, the dictionary-based approach is the most widely-used approach in CLIR. This is because dictionaries are increasingly available, there is wide variety in many languages, there is high recall and their approach is much easier to implement for query translation without expensive resources (i.e. sufficient training data for producing translations). Furthermore, the open nature of bilingual dictionaries makes their integration in ranking functions of weighting easy and particularly suit CLIR needs (Nie, 2010). These reasons, besides the fact that dictionaries have proven to be useful, result in making the use of bilingual dictionaries in CLIR is the most predominant techniques for query translation.

2.2.3.2 Machine Translation

Human languages are often ambiguous because words can be interpreted with different meanings. But fortunately, context in these languages is a rich resource that can used to reveal such ambiguity. Thus, the fundamental task of a Machine Translation (MT) approach is to deduce the statistical information, the knowledge resources, the collocations, the lexical rules and the syntactic information in contexts and then to use such information in order to find out the underlying meanings and the translation without any human involvement. Accordingly, machine translation systems can be utilized to translate a word, a sentence, a paragraph or a complete document. In CLIR, the MT systems can be utilized to translate

documents or/and queries and they may make the CLIR process much easier (Kishida, 2005) when they succeed to produce high quality translations.

Three types of MT system are known (Nie, 2010). These are the rule-based MT system, the statistical MT system and the hybrid systems that combine both of them. In the in rule-based MT systems, syntactic, morphological, semantic and lexical analysis and resources and manual-written linguistic hand-coding rules, i.e. grammar, are fundamental components. With respect to statistical MT systems they often provide the corresponding translations using the translation information and relationship obtained from *parallel texts* or *corpora* in two languages, aligned at some global level, typically sentence or paragraph. Parallel corpora, also known as bitexts (Resnik and Smith, 2003), are texts/collections that are composed from one language along with their equivalent translations in a different language.

Several models for statistical MT translation, known as the IBM models for translation, were developed by the IBM research team (Brown, et al., 1993). The basic idea is that if there is a source sentence a in $language_1$, the statistical model attempts to obtain the target sentence b in $language_2$ such that the probability is maximized (best translation of the source sentence a) This corresponds to:

$$Argmax_b\ pr(b|a) = argmax_b\ pr(\mathrm{b}|\mathrm{a}) \tag{2.26}$$

But, applying Bayes theorem this equation would result in:

$$Argmax_b\ pr(b|a) = argmax_b\ pr(a|b) * pr(b) \tag{2.27}$$

Where the conditional probability $pr(a|b)$ is the translation model which determines the translation probability from b to the sentence a and $pr(b)$ is a language model indicates the probabilistic mechanism for generating the sentence b in the target language, and is usually described by an N-gram model. Both a and b are segmented in consequent steps into segments smaller than sentences and for the estimation of $pr(a|b)$ it was assumed that there is a set X of possible alignments between the words in the two sentences (Nie, 2010). There are many details behind the statistical translation models (Brown, et.al, 1993; Manning and Schütze, 1999; Nie, 2010).

The IBM research group proposed five models namely, IBM model 1 to IBM model 5 (Brown, et.al, 1993). The models are used to assign the probability translation of a sentence b from a sentence a through an alignment process based on parallel corpora, as in such corpora various types of translation relationship between words in the source and target sentences can be extracted. For the purpose of implementing the IBM models, GIZA++ has been developed (Och and Ney, 2003). GIZA++ is a statistical MT toolkit that was designed for alignment in parallel corpora. GIZA++ is an extension of GIZA, which is part of a statistical MT toolkit named EGYPT, which is freely available software[5].

With respect to the CLIR task, the task of using the MT based approach is trivial. The user just needs to submit a query or document to translate and then the system would produce its translation as the aim is to generate good sentences in the target language.

[5] http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/GIZA++.html

However, MT systems still have some drawbacks and they are widely criticized. First of all, MT systems do not exist for many languages pairs and their resources are not available for many languages. Thus, finding sufficient parallel corpora for training the MT for such poor-resource languages is a major difficulty. Insufficient training data is not an easy drawback to overcome in MT systems and usually results in two major problems (Nie, et al., 1999). Firstly, it causes a MT system to fail in producing a translation for a given source term (OOV problem). This is especially true when queries include new technology terms, proper names and place names, etc. Secondly, such small training data often causes the MT system to produce chaotic translations that are inappropriate or invalid in the target sentence, although the context is used.

The invalid selection of a translation depends on the type of the MT system (Nie, 2010). On one hand, in statistical MT this is due to differences between the used parallel corpora for training the translation model of the MT system and the topic of the submitted query under translation. For example, the model may be trained on data extracted from the news domain, whereas the query to translate is in technical domain. On the other hand, rule-based MT systems fail to produce the appropriate translation due to the selection of the default translation word. Such a process of choosing one translation causes the IR system to limit their searching words to the original terms in the source query (Nie, et al., 1999); hence the valuable synonymy knowledge expansion is lost - thus, preventing the query, for example, from being expanded with these synonyms. Additionally, since the MT based approach depends on the adjacent context of the focal word to be translated, it may cause the statistical MT system to fail if the translation sense of the ambiguous word is not placed in the neighbouring or close words (Nie, 2010).

When the MT system is employed to the task of translating queries, ambiguity would likely occur (Pirkola, 1998). This is because the MT system is mostly based on interpreting words in the context and syntactic analysis as well (Navigli, 2009), which is usually lacking in queries due to their short lengths and due to the fact that users usually submit their queries in a non-well formed manner, not as a complete sentence in a syntactical form. It is not satisfactory to impose a burden on users so as to submit queries in a syntactical form. Similar trends were also concluded by Oard (1998), who showed that MT systems could obtain suitable performance if the submitted queries are long. The same study concluded, also, that MT-based document translation outperforms MT-based query translation. This is probably due to context utilization in documents. The same findings were also stated by Abusalah, et.al, (2005), who confirmed that machine translation-based approaches are more efficient to use with document translation since more context is available for disambiguation while dictionary-based methods are adequate for the query translation. Nevertheless, in his reasoning for the predominance of the query-based translation approach over document-based translation approach, Nie (2010) stated that it is not always that the document translation provides more benefits to query translation merely because MT systems usually translate sentences in documents separately and, thus, the rich context in documents is not always analyzed. Furthermore, Nie also stated that off-the shelf MT systems do not always satisfy the particular needs of CLIR and this is one of the major reasons for why the CLIR community turns to bilingual dictionaries. This is especially true if the translation is disambiguated. For instance, Ballesteros

and Croft (1998) showed that using MRD for query-based translation with some inexpensive methods such co-occurrence (discussed later) could achieve better performance.

2.2.3.3 Parallel and Comparable Corpora

Parallel corpora resources, which provide the same texts in more than one language, are often created by humans as a manual translation process with much effort or in some cases they may be acquired using software and MT methods. It is also prevalent to get parallel corpora from the Web, using a web crawler system (spiders are often utilized for crawling and collecting pages from the Web), as most organizations in non-English speaking countries often have the same content in their website in different languages. A typical example for parallel corpora is the UN articles. The UN often publishes its documents in a document repository in different official languages[6]. Examples also include the Holy Quran and the Bible, which are available in many languages even that are extinct.

There is a great interest in parallel corpora as a translation resource in CLIR. They have proven to be rich resources for translation although there are costly computations. This is because of their ability to provide good translations for new terms, technology, proper names and slang terms, especially when they are obtained from the Web. Furthermore, parallel corpora are beneficial sources for extracting linguistic knowledge such as POS-tagging and morphological analysis. Parallel corpora are also used to disambiguate translations when several alternatives are available for source terms (Nie, 2010).

For the purpose of translation, the majority of the approaches align firstly the parallel corpora into sentence or paragraph levels. The simplest approach of alignment is based on the assumption that parallel sentences, which are translations of each other, often would have a significant correlation represented in the same relative length but with the assumption that these aligned sentences would have the same order in the parallel corpora (Gale and Church, 1991). Such length can be measured in words, as in Brown's work (illustrated above), or in character, as in Gale and Church's work. For more details about the alignment process, refer to one of the mentioned studies, as the alignment process is relatively complex.

From CLIR prospective, these sentence-aligned pairs, which were extracted from parallel corpora, can be utilized for the purpose of training statistical models so as to align them into word level. The assumption that is made here is that the frequent appearance of two words together in parallel sentence pairs is an indication that these words are likely to be translations to each other. Thus, if correspondences at word level are identified, their estimated probabilities using the aligned-sentences can be obtained.

Apparently, there are much similarity between parallel corpora approach and MT. This is because in many cases the same translation model which is proposed for MT is often used to CLIR task on the basis of using parallel texts. However, among the proposed translation models of MT, the IBM Model 1 is the most widely used for the CLIR task (Kishida, 2005; Nie, 2010). The IBM Model 1 does not take into account positions of terms during the alignment and syntactic information is usually ignored.

[6] http://www.un.org/en/documents/index.shtml

Consequently, a word in a particular sentence can be a translation to any word in the aligned corresponding sentence. Nevertheless, the model suites the need of the CLIR task of producing most probable translations (Nie, 2003). Hence, using aligned sentences, the IBM Model 1 assigns translation probabilities that two words are translations for each other. In particular, the trained translation model on these parallel corpora provides a translation probability between a source language *a* and a target language *b* and/or vice-versa. The end product of the entire process is a bilingual term list with probabilities. Next, the automatically learned translation models are employed for the CLIR task. Such a translation model approach was followed in (Nie, et al., 1998; Nie, et al., 1999; Nie and Simard, 2001; Xu, et al., 2002). In some of these studies, well-formed parallel corpora, such as the UN collections, were used, while others employed the use of extracted parallel corpora from the Web. The latter is illustrated in the next section.

Using an English-French CLIR system with the TREC 6 CLIR test collection, in the study of Nie, et al., (1998) the translation models trained on the HANSARD, which is a parallel corpus in both English-French taken from the Canadian parliament proceedings[7], outperformed a MRD translation based approaches. The dictionary contained less than 8,000 words and all the possible translations were used, whereas the MT approaches, using high-quality MT systems, were slightly better than the translation models in terms of MAP (Mean Average Precision). But, it was observed that the used translations model, which was the IBM model 1, produced high probabilities for common translated words, which frequently co-occur with source words, instead of the production of such high probabilities to specific translated words. This drawback makes result biased towards those documents containing these common words and, thus, this drawback degrades performance of translation models. Accordingly, Nie and his team reported a strategy for incorporating the bilingual dictionary with the translation models but with the assumption that the suggested translations with dictionaries can be better adjusted with the use of the translation models according to the utilized parallel corpus. The results of such combination showed its positive impact on retrieval and it outperforms the effectiveness of the MT based approach. A similar trend and finding were also reported by a number of researchers (i.e. Kraaij, et al., 2003).

It is not always that a translation training model is used. Yang, et al., (1998) used pseudo relevant feedback to explore parallel corpora. The technique is simple and it works as follows: firstly, employ the source queries to retrieve documents in the source language. Secondly, retrieve the corresponding documents of the firstly retrieved set but in the second target language. Thirdly, extract the most frequent terms, which in turn are presumed to be implicit translations, from the latter set of documents and use them to retrieve the final result list in the target language. A somewhat similar technique was also presented by Davis (1998). In the study, the source English query term was used to search the English sub-collection in English-Spanish parallel corpora, which were aligned at document level. The same process was carried for the each translated Spanish term, which was obtained using a dictionary. This was done for every translation. Next, results sets of the retrieved documents were compared. The closest set among those retrieved by translations to those retrieved by a particular source term was considered as the best candidate translation. It is noted that the approach is useful for disambiguation

[7] http://www.parl.gc.ca/common/chamber.asp?Language=E

translation. A similar modified technique was also explored by Ballesteros and Croft (1998). The source queries in this study were used to search the Spanish part of the UN parallel corpora, but their corresponding English documents were also retrieved and the top ranked terms were extracted, using the Rocchio (Rocchio, 1971) approach, and ranked according to their scores. Thus, based in the best scores, their translations were selected as the possible candidates.

Other approaches investigated the use of parallel corpora with LSI (Latent Semantic Indexing)(Landauer and Littman, 1990). Basically, LSI is applied to a matrix of terms by documents. Hence, words in the collection vocabulary are defined as vectors and have a dimension equal to the total number of distinct words in the collection. However, in such a representation matrix of terms by documents, the matrix will be very sparse and large and computationally costly to manipulate (Manning and Schütze, 1999). To avoid such overhead and to minimize such representation to a workable, size Singular Value Decomposition (SVD) is used.

The use of the LSI has been extended to CLIR using parallel corpora (Landauer and Littman, 1990; Mori et al., 2001) so as to create a multilingual, language independent and multi-dimensional indexing space and to provide a utility for implicit translation of queries. The basic idea is to combine the parallel corpora, which is aligned at document level, into new documents that contain terms in both languages. Then, the SVD technique is used to map the sparse term-document matrix into a reduced semantic space that contains terms vectors in both languages. In this semantic space, close words would likely have similar or close representations. Queries, regardless of their language, can be also represented in the same generated semantic space, by vectors, and their scores in documents can be computed using the cosine similarity. As stated by Nie (2010) no explicit translation for query or document is needed. Mori et al. (2001) implemented such an approach using the NTCIR-2 English-Japanese test collection. However, in his analysis for the use of the LSI plus the results that were obtained by Mori and his team, Nie (2010) summarized two major difficulties for using LSI in CLIR. Firstly, it was shown that using LSI is computationally expensive which causes the use of small parallel corpora in spite of the fact that the utilization of the parallel corpora approach in CLIR depends solely on using large training data acquired from these parallel corpora. Secondly, LSI didn't yield competitive results to most approaches used in typical experiments on large test collections.

Although parallel corpora had proven to be valuable CLIR experiments, they have their drawbacks. Firstly, such parallel corpora are not available for many languages or they are not large enough for estimating accurate translations and, hence, resulting in the impractical use for many languages. Secondly, parallel corpora depend solely on the restricted parallelism between language sentences (Nie, 2010). This feature does not always hold in many corpora. Thirdly, most available corpora are in a specific field (i.e. the UN collection). This fact makes the learning process of translation susceptible to failure in disambiguating terms with fine-grained translations, if the domain of the collection being searched is different from the domain of the collection that was used in the training process (Monz and Dorr, 2005).

Comparable corpora were also investigated as sources of translations (Sheridan, et al., 1998; Molina-Salgado, et al. 2002). As in parallel corpora, comparable corpora are also texts in more than one

language but the text in each language does not present as the exact translation to each other, but it is topically similar (cover the same contents). Comparable corpora are more available than parallel corpora (Nie, 2010) especially in news articles, which are usually published on the same global and local events. Approaches that are used to extract translations from parallel corpora are different from those utilized in comparable ones, due to the noise in the latter corpora. Many studies employed the use of a similarity co-occurrence measure or an association degree between two words across languages within comparable texts so as to construct a similarity thesaurus (Sheridan, et al., 1998; Molina-Salgado, et al. 2002). Thus, words that co-occur frequently and concurrently in comparable texts would likely be similar and, hence, they can be valid entries in the similarity thesaurus. Thus, the weighted translations from the thesaurus are used instead of the source query terms and/or in terms of query expansion. Results in most studies that utilized the use of comparable corpora, e.g. Molina-Salgado, et al. 2002, Resnik, 1998 and Nie, et al., 1999 reported that the use of comparable corpora is beneficial for languages with no parallel corpora, although comparable corpora based approaches may introduce much noise. As comparable corpora are often extracted from the Web, more details about their use are provided in the next sub-section.

2.2.3.4 Utilization of the Web

Motivated by the fact that the Web is very diverse in different types of knowledge in many languages and bilingual resources, and that MT systems and parallel/comparable corpora are relatively few for a number of languages, in recent years there is a great interest in employing the Web as a rich resource for translation. A number of studies (Resnik, 1998; Nie, et al., 1999; Resnik and Smith, 2003) have investigated methods to mine the Web for the purpose of building bilingual translation resources (i.e. comparable corpora). The key idea is to mine, compare and exploit the common structure in bilingual websites using several forms of evidence so as to align these pages. An example of such evidence is the assumption that parallel pages are often named using the same name, but only a different small part that indicates content language (Nie, et al., 1999). Another example for the basis of pairing websites is that parallel Web pages often express similar topics and consequently they tend to have close lengths. After the parallel pages are obtained, the constructed Web corpora are employed to create a statistical translation model, as illustrated in the parallel corpora section, which in turn will be used for translating queries.

In the study of Nie and his colleagues, such a technique was compared with several other approaches for translation using both TREC6 and TREC-7 in both English-French and French-English CLIR. Results obtained in the study showed that translation models based on parallel extraction of pages using such techniques, when they were combined with dictionaries as in Nie, et al., (1998), which was shown in the parallel and comparable corpora section, are globally comparable to those obtained using one of the best MT systems. In the comparison with the translation model, which was based on the well aligned Canadian HANSARD corpus with the same combination with dictionaries, results revealed that both the translation that was based on the extracted Web corpus and that was based on TREC-6 yielded similar performance, whereas it was less effective in TREC-7. As stated by the developers, this difference in

performance is based on the fact that the queries in TREC-7 contain many names of countries and regions, which were not well translated using the translation model that was trained on the extracted bilingual corpora from the Web. Automatically extracted Web corpora were also used in the Nie and Simard (2001) study, which is covered in section 2.2.4.2.5.

Extending the work of Resnik (1998), Resnik and Smith (2003) proposed architecture for mining the Web in order to construct parallel corpora. In that architecture, which was developed as software called STRAND (Structural Translation Recognition, Acquiring Natural Data), the structure of the pages that contains the same links of the same documents in different languages are used. In that way, pages with the same contents are considered as parallel. Results reported in this work show the effectiveness of the STRAND architecture in building parallel corpora for a language pair with no easy ways to obtain parallel corpora for them such as English and Arabic as well as its effectiveness in general for creating parallel corpora. However, the UN collection was not released yet during the time of this study.

2.2.4 Significant Difficulties during Translation

It was previously discussed that the bilingual MRD approach for query translation is the most dominant approach in CLIR as they are abundant, readily-available in many several languages, easier to acquire and provide us with enough recall. Nevertheless, the approach identifies two major difficulties, which can affect CLIR retrieval - these are words that are not covered by dictionaries and translation ambiguity (Hull and Grefenstette, 1996).

The problem of OOV arises from that fact that that some terms during translation may be not covered and outside the scope of a certain translation resource. The OOV is a widespread problem in CLIR. OOV terms are often inflected words, proper nouns, spelling variants, cross-linguistic names and technical terms (Pirkola, et al., 2001). The problem is that missing the translation of such OOV words, which are often major keys in searching, degrades performance of CLIR systems. The OOV problem is related to the above question in the introduction of this section, which is: which translation source and approach can be used/acquired so as to perform the translation process?

Additionally, whenever a translation resource is used, a bilingual MRD for example, it is often that several possible translation candidates are obtained for a particular source word. This is because words are often polysemous (several meanings), as was stated above. In such a case the word is called ambiguous. Translation ambiguity is one of the most challenging problems in CLIR. The problem of translation disambiguation attempts to answer the third question in the introduction of this section when crossing language barriers, that is: how should a translation be utilized in the CLIR weighting and retrieval process?

The next two sections illustrate some of the proposed approaches to these two major problems.

2.2.4.1 Resolution of Terms Coverage Problem: Out-Of-Vocabulary

The OOV problem is severe and could result in a significant negative impact on retrieval effectiveness. For instance, Larkey, et al. (2003) showed that the functioning performance of the CLIR system degrades by more than 50% in terms of average precision if proper nouns and named entities in the source queries are untranslatable. Since bilingual dictionaries are the most popular translation resources in CLIR and the only available option for many languages with few resources, untranslatable terms are fairly prevalent in this translation approach.

It was previously shown that the OOV terms may result for different reasons (i.e. proper nouns). However, English, as a primary language in the Web and the most used language in CLIR systems, contributes to the OOV problem because of the short vowels of English words which can be written in different forms of phonetics in other languages, especially those that are inflectional (i.e. Arabic and Hebrew). For instance, according to AbdulJaleel and Larkey (2003), Whitaker found 32 different English spellings for the name of the ex-Libyan leader Muammar Gaddafi, which is originally an Arabic proper name. However, such spelling variants due to vowel letters in the English language.

Numerous approaches have been explored to provide a solution to the OOV problem, e.g. the use of domain-specific and special dictionaries and backoff translations, as illustrated earlier. Nevertheless, there is still an exception that some words cannot easily be translated. Hence, other approaches for solving the problem were proposed. Among them, the employment of the Web and the transliteration technique are discussed.

2.2.4.1.1 Exploring the Web

Besides the employment of the Web to build bilingual parallel/comparable corpora, the Web is also becoming a viable resource to translate OOV terms, in general, and up-to-date terminologies, in particular (Kishida, 2005). Basically, the use of the Web for resolving OOV is divided into two major approaches that are the use of the bilingual search-results snippets (query-biased summary) in mixed documents and/or the use of hyperlinks and anchor texts.

One such approach when search-engine result snippets are used is the utilization of the co-occurrence of terms in both the source and target languages in non-English texts, as in Arabic and Chinese. As was previously discussed that texts of such non-English languages often contain several embedded words in a different language, mostly in English. Zhang and Vines (2004) stated that in Chinese Web pages, English terms are very likely to be the translations of their immediately preceding Chinese terms. Furthermore, co-occurrence of bilingual pair terms often follows a particular pattern that could be used to extract these English translations (i.e. English translation in Chinese pages is often placed into a parenthesis). According to this observation, Zhang and Vines (2004) and Zhang, et al. (2005) proposed a technique based on such co-occurring terms in Chinese and English pairs. In the technique, pages that contain such co-occurrence of terms are retrieved using search engine APIs (Application Programming Interface) and the translations that follow the pattern in the snippets of the top retrieved documents are extracted based

on the frequency of a certain pair. When there is more than one translation available, the developers used a Markov Model (HMM) based co-occurrence statistic to extract the most common translation of the OOV term. Using some NTCIR collections results showed that the approach had a significant impact on improvement as many unknown terms for the Chinese-English CLIR experiment were found in such a way. Li, et al. (2009) used a similar approach but, with the addition of translations of the related terms in the query. Such a technique may result in eliminating noisy translations.

Translations of OOV terms can be also obtained using hyperlinks and anchor texts in Web pages (Cheng, et al., 2004; Lu, et al., 2004). Usually anchor texts are used, in terms of descriptions for examples, to indicate which parallel page is linked to which Web page. Thus, different anchor texts in multiple languages might link to the same pages. Accordingly, such anchor texts were used to build a parallel corpus of anchor text sets. Following this, the translation candidates for each query term are extracted from anchor text sets, which contains the query term. Next, a probability is computed between the query term and each translation candidate that co-occurs with that query term in the same anchor text sets. Hence, a translation candidate that frequently co-occurs with the query term in the same anchor text corpus would likely obtain higher probability. Results reported in this work showed that this approach improves the effectiveness of CLIR performance

It important to note that translation of OOV terms using the Web is usually utilized in a complementary role with the bilingual dictionaries. Nie (2010) stated that the use of such an approach could improve retrieval performance significantly.

2.2.4.1.2 Transliteration

One of the approaches used to overcome the problem of OOV is to phonetically transliterate unknown terms. Transliteration means to represent a word in a particular language in the closest corresponding letters in a way that the pronunciation becomes close as much as possible across languages (AbdulJaleel and Larkey, 2003). Thus, the problem of transliteration is, in fact, how to identify phonetic correspondences between languages that are different in their orthographic structure (Zhou, et al., 2008) and how to produce rules representing characters and how those characters are orthographically mapped to their corresponding characters. This is especially difficult as each language may have different phonemes with no equivalence in others languages. Once such rules are available, they can be immediately used to transliterate the OOV terms.

Simple approaches that are based on orthographically mapped substrings between languages were used. Such approaches are useful for languages that share similar alphabets, such as English and French. For instance, sometimes French proper nouns are left in during translation as they appear or they can be considered as misspelled words that need a few mappings of letters (Buckley, et al., 1998). For many pairs of languages, however, transliteration is relatively challenging, when these language pairs use different alphabets, such as in the case of English and Arabic. In such a case an extra process, known as phonetic mapping (Zhou, et al., 2008), is needed.

Phonetic mapping is primarily performed using tables, for example, extracted from parallel corpora, particularly, paired lists of aligned terms in both the source and target languages (AbdulJaleel and Larkey, 2003; Zhou, et al., 2008). Once such aligned lists are made available, then somewhat similar techniques to those used in statistical MT systems are used for phonetic mapping. For example, a character-based alignment of the paired lists can be employed to estimate a transliteration probability for each phonetic representation sequence in a source language with its corresponding sequence in a target language. This is the automatic training phase, in which a table of transliteration probabilities is identified for the phonetic representation. Thus, to produce transliterations for a source word, the word is segmented according to its corresponding segments in the generated phonetic representation table. Next, all the possible transliterations are obtained for each segment and, thus, the possible transliterations for the source word with their probabilities to be the correct words can be obtained. An example for a typical statistical technique developed for transliterating English to Arabic in CLIR is provided in section 3.5.2 in the next chapter.

2.2.4.2 Translation Disambiguation and Weighting Difficulty

It was discussed in section 2.2.4 that a word can be ambiguous. To disambiguate a word, two primitive methods can be followed. These are the explicit disambiguation approach (translation selection) and the implicit disambiguation approach (no translation selection). The first approach is based on the intuition that the most frequent translation in dictionaries is often listed first and, thus, one might use only the first matching translation. Such a naïve approach is simple and it disambiguates translation explicitly as it chooses only one word. But, the approach has two major limitations. First, it is not always correct that the first instance in a bilingual dictionary is the best translation (i.e. if there are many possible candidate translations, dictionaries may order them alphabetically). This phenomenon invalidates the fundamental assumption of the approach. Second, such elimination of translation alternatives of query terms would prevent the CLIR system from retrieving many potentially relevant documents, although at the same time many superfluous translations would likely be excluded.

A second approach (no translation selection) is to replace each term in the query with all its possible translations, known as the unbalanced query (Oard, 1998; Levow, et. al, 2005). This is an undesirable trait in IR processes because it would likely bias the retrieval list towards those source terms with more possible translation candidates, instead of those with very specific translations (a few number of translations) as their contributions in weighting would be less.

Both approaches are not good enough to perform a sophisticated CLIR task (Ballesteros and Croft, 1997) so, other approaches were proposed. The first approach attempts explicitly to select the best translation only, according to somewhat observed phenomenon (i.e. high frequently co-occurrence between a source term and one of its translation candidates in a corpus), while the second approach attempts either to re-weight query terms, based on their matched translations in a dictionary for example, when no translation probability knowledge is available or it attempts to estimate a probability for each possible translation candidate (when such probabilistic knowledge is available) for a certain level of preference, according to

some statistics, e.g., the frequency of appearance of a particular translation in several dictionaries. Next, these estimated probabilities are used in weighting their corresponding translations. In the second approach, all or some of the translations are used, unlike the first approach which mostly chooses only the best translation. Furthermore, the estimated probabilities or re-computed weights of translations are used implicitly during retrieval. Indeed there are some other approaches to translation disambiguation such as query expansion and POS approaches.

The next section discusses some of the proposed techniques for translation disambiguation. Firstly, balanced translation queries and structured query model and some of its variants are presented, such as bidirectional translation models and those approaches that combine probabilities with IDF. Some of these approaches are used when no translation probability knowledge is available, some are employed when such probability knowledge is obtainable while others (like bidirectional approach) can be used if such probability knowledge is available or not. Secondly, translation disambiguation using co-occurrence of terms in monolingual and unlinked corpora is discussed, as an example for those approaches that choose best translation. Finally, the query expansion technique for translation disambiguation is illustrated as an example of the other approaches. It is important to note that a particular approach for translation disambiguity may depend solely on the utilized resource(s) for translation. For example, the structured query model approach was proposed for term weighting difficulty when translations are obtained from a dictionary, while a variant of the same model (i.e. probabilistic structure query model) may estimate probabilities based on parallel corpora and/or dictionaries.

2.2.4.2.1 Balanced Translation Query

It was shown that a source query term with many possible translations, mostly common terms, would likely result in a biased result list, as terms with few translations, mostly specific, will not compete. This is especially true when a bag-of-words retrieval model, in which contributions of terms in weighting are handled independently, is utilized, e.g., the vector space model.

In order to suppress the contribution impact of such general terms over specific terms, Levow and Oard (2002) proposed a re-balancing mechanism for translations of terms. The rebalancing is obtained by averaging the weights of all the corresponding translations of a given term. Given a term query q_i with a set of translations $T(q_i)$ with t representing elements in this set, the weight of the query term q_i in document d_k, denoted as $wt(q_i, d_k)$, can be averaged as (symbols are derived from Levow et al. (2005)):

$$wt(q_i, d_k) = \frac{\sum_{\{t|t \in T(q_i)\}} wt(t, d_k)}{\{t|t \in T(q_i)\}} \tag{2.28}$$

$$wt(t, d_k) = Ret\_Mod\_Funt(TF(t, d_k), DF_t, L_{d_k}) \tag{2.29}$$

Where $wt(t, d_k)$ is the weight of the translation t in document d_k, $TF(t, d_k)$ is the frequency of the translation t in document d_k, df_t is the number of documents that contains the translation t, L_{d_k} is the length of document d_k (number of terms that occur in the document) and *Ret_Mod_Funt* is a term

weighting function that is increasing in *TF* and is decreasing in both *DF* and *L*. Averaging all translations would give more weights to scarce translations, and thus their documents would be favoured, as rare translations are not always the correct ones (Levow, et.al, 2005). Based on this drawback, other approaches, e.g., probabilistic structured query(discussed later) are proposed.

2.2.4.2.2 Structured Query Translation Model

The structured query model was initially developed for monolingual retrieval so as to expand queries using thesaurus. The key idea behind the structured query model is to treat all the listed terms that are obtained from a monolingual resource in a given language as if they are synonyms or instances of a particular term, and consequently they will have the same impact as the stemming process (Darwish and Oard, 2003a, 2003b) on both TF and DF components. The structured query model is included in the InQuery retrieval system under the synonym operator (#SYN) (Broglio, et al., 1994). Both Ballesteros and Croft (1998) and Pirkola (1998) utilized the same operator to fit the CLIR process, in what is called the Structured Query translation Model (SQM), for the purpose of handling all the candidate translations of a query term as synonyms in the target language. Assume that a query term q_i in a source language has some known translation candidates, then using #SYN operator/SQM, all these alternatives would be treated as synonyms in the target language. The impact primarily appears in the weight computations of the query term q_i. Given the above arguments with the assumption that $T(q_i)$ is the set of the known translations, with t representing elements in this set and all elements handled as synonyms, for the query term q_i, the structured query model estimates *TF*, *DF* and document length as follows:

$$TF(q_i, d_k) = \sum_{\{t|t \in T(q_i)\}} TF(t, d_k) \tag{2.30}$$

$$DF(q_i) = \left| \bigcup_{\{t|t \in T(q_i)\}} \{d_t\} \right| \tag{2.31}$$

$$L'_{d_k} = l_{d_k} \tag{2.32}$$

Where $TF(q_i, d_k)$ is the term frequency of the query term q_i in document d_k and $DF(q_i)$ is the number of documents in which the term q_i occurs, $TF(t, d_k)$ is the term frequency of the translation t in document d_k, d_t is the set of documents containing the translation t, L'_{d_k} and l_{d_k} is the length of the document d_k (the length of a document is kept in the SQM). The effect of structuring the query term is a new expanded query, whose TF and DF computations are re-computed.

Pirkola experimentally showed that the proposed structure for queries with both general and technical dictionaries is effective and could achieve the same level of performance when using a monolingual IR system, using Finnish queries, which were translated into English by a dictionary, against the TREC English collection. In other studies (Pikola, et al., 2001; Pirkola, et al., 2003) the same results were also concluded. Additionally, it was shown that synonyms-based structuring, in which the translations of a certain term query are grouped together, outperformed compound-based structuring. Compound-based

structuring is often obtained by merging each translation sense of the first word in the compound with e all the translations that are obtained by the second word in the same compound. When all possible combinations are obtained, they would be handled as synonymous compounds, resulting in compound-based structuring. This is exactly the (#UWN) operator in the InQuery IR system, which attempts to find any order for the words listed in its arguments in a window of size N.

For its simplicity and its need for only cheap resources, which are often dictionaries, structured query, which is called Pirkola's model throughout this thesis, is becoming one of the widely used approaches in translation disambiguation. In fact, structured query does not disambiguate translation explicitly but the disambiguation takes place implicitly and, thus, SQM has the same impact of translation disambiguation (Kishida, 2005).

Many studies derived several variants using the same technique. Stated in Darwish and Oard (2003a), Kwok presented a variant to structured query by substituting the union operator with a sum in order to make the implementation details less complex:

$$DF(q_i) = \sum_{\{t|t \in T(q_i)\}} DF(t) \tag{2.33}$$

In the same study, Darwish and Oard proposed another variant for this formula:

$$DF(q_i) = \underset{\{t|t \in T(q_i)\}}{MAX}[DF(t)] \tag{2.34}$$

Results reported that there were no significant differences when these two variants were compared to the union in Pirkola's method. Another variant of the SQM that had been introduced is the Probabilistic Structured Query (PSQ), as will be seen next.

2.2.4.2.3 Probabilistic Structured Query Translation Model

Pirkola's structured query model has a potential drawback. Since all translations are treated as equally likely, the overall document frequency of the query term (actually its translations for monolingual retrieval) would be high, if the document frequency of one of these translations is high, too and thus, resulting in a low weight for the corresponding query term, although there may be very specific translations among the possible candidate with low document frequency. Because of this drawback, Darwish and Oard (2003a, 2003b) proposed a variant method of Pirkola's structured query, known as the Probabilistic Structured Query (PSQ). Besides the aim of resolving the illustrated drawback, the assumption behind the PSQ model is that using statistical models usually result in translations with both strong and weak probabilities. Hence, incorporation of these probabilities in the term frequency and the document frequency computations of Pirkola's model, documents with specific translations, which probably correspond to strong probabilities, would be ranked at the top.

Probabilities of translations can be estimated from aligned parallel corpora or dictionaries, in which translations are ordered by their frequency of use. It is possible also to obtain the estimated probabilities using frequency distributions of translations in multiple sources, as was done by the developers of the

approach. As translation probabilities (evidences) are obtained, they are combined into the weights so as to contribute to the *TF* and the *DF* of a given term as follows:

$$Weighted\_TF(q_i,d_k) = \sum_{\{t|t\in T(q_i)\}} TF(t,d_k) \times pr(t|q_i) \quad (2.35)$$

$$Weighted\_DF(q_i) = \sum_{\{t|t\in T(q_i)\}} DF(t) \times pr(t|q_i) \quad (2.36)$$

Where $pr(t|q_i)$ is the probability of the query term q_i to be translated to the translation t. The approach causes documents that contain the most likely translation to be retrieved higher than a document with a superfluous translation. Using the TREC 2002 Arabic collection, Darwish and Oard showed that their probabilistic approach yielded better performance effectiveness, when it was compared to both Kwok's variant of SQM and the *MAX DF* method, which were illustrated above. The same conclusion was also reported when results were compared to the structured query model.

2.2.4.2.5 Bidirectional Translation Disambiguation

Instead of a unidirectional translation process, bidirectional translations or two direction translations (Boughanem, et.al, 2002), in which translations are executed in both directions from a source language to a target language and vice versa, were also investigated. The hypothesis here is that if the set of equivalent senses for a source term is backward-translated term by term into the source language, using dictionaries for example, the preferred translation is then the target word, whose set of equivalent translations into the source language contain the original source term.

Aljlayl, et al (2002) employed a similar approach in Arabic-English CLIR experiments. Results reported in this work showed that the performance was statistically significant when the bi-directional translation approach was utilized.

It was shown previously that using statistical translation models may result in the preference of general translations, which are basically not stopwords, rather than specific translations, as high probabilities are often produced for general translations, as it was discussed in the parallel and comparable corpora section. Therefore, one of the proposed approaches to handle such a problem is to combine statistical models with a bidirectional approach (Nie and Simard, 2001). The underlying assumption is that when a bidirectional translation is employed, precise translations tend to backward-translate to the source term with somewhat highly probability, unlike general and extraneous translations, which are expected to produce many other source-based translations with low probabilities. In Nie and Simard's work, bidirectional translation probabilities were computed as the multiplication of the probabilities in each direction. The translation models were trained from a set of parallel Web pages that are automatically obtained from the Web. Nie and Simard concluded that the use of this approach did not imply an improvement over unidirectional translation. The same approach of using bidirectional translations with statistical models was also followed by Wang and Oard (2006), who particularly generalized the probabilistic structured query in a way that the translation direction was not determined. The developers

called such an approach the meaning matching model, as for the language of the query's terms, there may be some synonyms in the document language, which may share the same meaning, and vice-versa. The meaning matching model was expressed in the study as sets of synonymous translations in both directions and synonyms terms that are derived using synonymy resources (i.e. EuroWordNet) along with their probabilities, which are derived from statistical models based on sentence-aligned parallel corpora extracted from news. Then the two probabilities were combined in a statistical synonym model. In the work of Wand and Oard, the probability that both the terms *s* and *t* share the same meaning, denoted as *pr (s ↔ t)*, was computed as follows:

$$pr\,(s \leftrightarrow t) = \sum_{m_i} pr(m_i \mid s) \times pr(m_i \mid t) \tag{2.37}$$

Where $pr(m_i \mid s)$ is the probability that the term *s* has the meaning m_i while $p(m_i \mid t)$ is the probability that the term *t* has the meaning m_i and both *s* and *t* are in different languages. The meaning matching probabilities are then incorporated in the *TF* and *DF* weighting in a similar way to the PSQ. Results reported in this work showed that the proposed method could yield better performance than using only unidirectional translation knowledge and comparable effectiveness to monolingual retrieval using the same experimental setup. Bidirectional translation can be also performed on document level, instead of word level, as was shown by McCarley, (1999), which was illustrated in previously.

2.2.4.2.6 Disambiguation with Translation Models and IDF

It was shown in section 2.2.3.3 that a translation model may yield high probabilities for common translations, as such common words usually result in more translations than specific terms, and thus their weights and significance are increased. Because of this drawback, Nie and Simard (2001), and Nie (1999) attempted another strategy. The technique merged each translation probability, which was provided by a statistical model trained on parallel Web corpora aligned at sentence level, with the inverse document frequency of the translation word by multiplication. This is usually done after removing those translations with low probabilities. The basic assumption here is that the strongest translations (whose probabilities are higher than a certain threshold) may result in lowering probabilities of generic words, whereas probabilities of specific words increase. Results showed that the approach is beneficial and contributes to performance effectiveness.

2.2.4.2.7 Translation Selection approaches

It was shown is section 2.2.3.3 (parallel and comparable corpora for translation) that parallel corpora can be used for translation selection, as in the studies of Yang, et al., (1998), Davis (1998) and Ballesteros and Croft (1998), which were illustrated above. However, it was also illustrated that such parallel corpora are not available for many languages. Therefore, a considerable number of studies in

CLIR explored the efficiency of disambiguating translations, in terms of translation selection, using the frequency statistics of simultaneous appearance of paired terms in unlinked target corpora. Unlike Word Sense Disambiguation (WSD) and parallel or comparable corpora, translation disambiguation using co-occurrence is cheap and it does not require tremendous effort because its statistics are often computed from a monolingual and unlinked collection, possibly the corpus that is being searched, rather than parallel corpora.

Disambiguation Using Co-occurrence Statistics in Unlinked Corpora

Co-occurrence techniques are based on the hypothesis that correct translations tend to co-occur together in the target language collection (Ballesteros and Croft, 1998). Therefore, the valid translation among a set of possible synonymous candidates of a certain source query term is expected to have high frequency of co-occurrence with the translations of the other terms in the same source query. In such cases, the problem becomes how to estimate the strength of association (the degree of similarity) between each paired element in the produced set. This is the co-occurrence problem.

Different similarities measures (degree of association) can be used to measure how frequently two terms co-occur in a predefined window, for example. However, the basic common used similarity measure is the Mutual Information (MI) (Church and Hanks, 1990), which estimates the probability for the number of occurrence times in which a term *a* co-occur with another term *b* in a window with fixed size of N (i.e. N= 50). Based on a variant of MI co-occurrence, Ballesteros and Croft (1997, 1998) developed a method for disambiguating term translation and phrase translation when a phrase is translated using a dictionary in a word-by-word fashion. Results showed that the approach achieved significant improvement in performance and in effectiveness of phrasal translation as well, although the approach employed only a monolingual corpus. Jang, et al., (1999) followed a similar approach by employing also the MI and obtained similar results, also, in a Korean-English CLIR task with TREC-6. However, the distinction of Chang's work is that it eliminated the number of combination of translations to only those in order in the query. Hence, if the query consists of three words ABC in that order, only AB and BC are considered, but not AC.

If the query is long or it contains many words, it would be inadequate to carry out much co-occurrence computations, as it would be computationally expensive (Kishida, 2005). Therefore, some studies attempts to minimize this limitation. Adriani and Rijsbergen (2000), Gao, et al. (2000) and Gao, et al. (2001) proposed methods based on the cohesion of each translation a_i in the set of all possible translations A of a particular source term with each set B_m, which is the set of translation alternatives for other source query terms, but only the maximum similarity is selected between each translation a_i and the translations in each set B_m. Next, the cohesion is computed as the sum of all similarities and the cohesion in A is chosen as the best translation. In Gao's work the used similarity measure was a point-wise mutual information, while Adriani and Rijsbergen, used the Dice coefficient similarity. Later, Gao, et al. (2002) extended his above-mentioned study by identifying another factor that has an impact on computing co-occurrence significance. Since terms at distant positions in previous studies tend to be

treated as if they are closer to each other, Gao introduced a 'decaying co-occurrence model', which is based on the distance factor between a pair of terms. Results reported on experiments using the TREC-9 collection showed that the proposed approach is better than the basic MI model.

Some researchers also assumed that the association among possible translations can be presented as a graph (Monz and Dorr, 2005; Zhou, et al., 2008), in which nodes represent the translation candidates for the various source terms and edges between each two nodes represent the assigned weights, which is computed with the specific co-occurrence similarity measure and hence, weights determine the strength of the link between every possible pair of translations. Next, all weights are then used to derive a global decision about the importance of a translation candidate. Hence, the cohesion is computed from the assigned weights to edges dynamically (Zhou et. al, 2008). As stated by Nie (2010), this is different from previous approaches, which are based on using static cohesion for similarity measures of query translation. Furthermore, results in Zhou's study revealed that there was no remarkable improvement for the dynamic selection approach over static approach. However, in Monz and Dorr's work, which was an English-German CLIR retrieval, results showed significant improvement, but over the dictionary based approach when it was implemented without any term weighting.

2.2.4.2.8 Other Approaches

There are still other approaches for translation disambiguation. As an example, query expansion for disambiguation is discussed.

Translation Disambiguation Using Query Expansion

In monolingual retrieval, the intended information needs during the IR process may not be fully expressed by user's queries and hence some relevant documents may not be retrieved. For this cause, query expansion techniques were proposed in order to enlarge/enrich queries by adding some useful terms related to the original queries and, thus, their coverage is expanded (Rocchio, 1971). The type of the used resources to expand queries has a major impact on this step. In fact, different techniques in query expansion are distinguished from each other by the sources that they utilize for gathering the related terms. Types of exploited resources include: explicit synonymous terms from thesauri, related queries extracted from query logs and clickthrough data, blind relevance feedback and results from search engines. Among these approaches, the Pseudo Relevance Feedback, known also as blind feedback, is one of the most widely used in CLIR. The blind feedback expands queries by some related terms, which are extracted from some relevant documents in the initial retrieved list. Such relevancy can be obtained from the query issuer in terms of explicit feedback, by clicking a typical "like-this-document" button on a certain document, or the feedback can be implicitly indicated, by marking top documents as relevant. Alternatively, the expansion process can be performed by choosing the most significant terms from the top N documents. Experimental evidence, as in Buckley, et al. (1995) showed that pseudo relevance feedback is widely accepted as an effective technique for improving performance in IR.

In CLIR, query expansion can be considered as a corpus-based approach for translation disambiguation. But queries can be expanded either before they are being translated (pre-translation feedback) or after the translation (post-translation feedback) (Ballesteros and Croft, 1997; McNamee and Mayfield, 2002a). As the name indicates, in the pre-translation feedback approach the expansion process of source queries is performed from documents in the source language, if any is available, prior to translation. The post-translation feedback translates the query first and then sends it to a local document collection in the target language so as to extract terms for expanding the translated query. Many studies explored the impact of both types of feedbacks. Ballesteros and Croft (1997) showed that the combination of both pre-translation and post-translation query expansion together improved both precision and recall and it is more effective than each of pre-translation or post-translation expansion alone. This is because the pre-translation approach generates a robust foundation for the followed translation, especially for short queries, resulting in the improvement in both precision and recall, whereas the post-translation feedback eliminates irrelevant translations, resulting in minimizing translation ambiguity and hence, resulting in recall improvement. However, McNamee and Mayfield (2002a) showed that if the source query contains only too few words, post-translation query expansion provides only little improvement. Accordingly, the pre-translation approach is better than the post-translation. But the effect of the used query expansion technique is solely based on the used document collection that is employed for the extraction of terms (Nie, 2010).

Document expansion had been also explored Levow and Oard (2000). The assumption that was made is that a document usually contains only a part of the query terms and, thus, if the documents are used, using some techniques, as queries, this may improve retrieval effectiveness. A general technique for such document expansion firstly extracts the most significant terms from the retrieved set of documents by a particular query (document in this case). Next, the queries (documents) are expended with these significant extracted terms, resulting in a new expanded document collection. In Levow and Oard (2000), documents were firstly translated in the source language and each translated document was being used as a query. Significant terms are then extracted from top documents and used to expand the translated documents. A similar approach was also followed by Levow, et al. (2005). However, results also showed that no significant improvement was obtained in the retrieval effectiveness. In Darwish and Oard (2003b), who used a slightly different expansion for documents but, within the same context of creating expanded document collection, similar findings were also obtained.

2.3 Traditional Multilingual Information Retrieval

In CLIR the retrieval task is bilingual because all the target documents are written in a single language. It is possible, however, to have a multilingual document collection in which several monolingual document collections, each of which is in a single language, are presented (Lin and Chen, 2003; Chen and Gey, 2004b; Luo, et. al, 2008). In such a case, in which these corpora are not parallel, the bilingual retrieval task becomes multilingual. This is called Multilingual Information Retrieval (MLIR). In particular, MLIR is the task of searching and retrieving relevant documents into a single unified ranked list in different

target languages, using a query written in a single source language. Two major approaches are used for traditional MLIR indexing: the centralized architecture and the distributed architecture.

2.3.1 Centralized Architecture

Since the multilingual collection contains at least two languages, the first approach, which is the centralized architecture/indexing, puts all documents, regardless of their language into a single centralized index (Nie and Jin, 2003; Gey, et al., 2001). Queries are translated into all the document target languages and a set of several monolingual queries in different languages is obtained. Next, for each single query, its different translated versions, including the source, are concatenated together to form a single big query, which is submitted to the single multilingual collection. This is the dominant approach in the centralized architecture.

Instead of this approach in the centralized architecture, an alternative is to translate the various languages presented in documents into the source query, employing some kind of fast translation such as using bilingual word-lists created by translated document words using MT systems, as was discussed in section 2.2.1 (query translation versus document translation). All the translated documents in the source language of queries are then indexed together into a single index. Hence, the source query is used to search this big index, monolingually. This approach is taken by Chen and Gey (2004b).

The strength of the centralized approach comes from the use of a single index. This is due to two reasons. First, using a single index makes weights of terms, regardless of their languages, become more comparable because they are retrieved by a single IR model and computed in the same way (Nie and Jin, 2003). Second, the centralized architecture avoids the problematic merging difficulty (see next section). Nevertheless, the centralized index in multilingual or bilingual retrieval task has a major drawback in that index terms weights are often over-weighted (Lin and Chen, 2003). This is because the number of documents (DF) for a term increases, while the number of occurrences of a term (TF) is kept unchanged and thus terms are over-weighted. Consider a multilingual collection containing 6,000 monolingual Arabic documents along with 70,000 documents in English. When all these documented are placed together into a single collection, the N value (number of all documents) in the IDF factor of terms in standard weighting schemes, which is computed as log(N/DF), for the Arabic sub-collection will increase significantly to 76,000, instead of 6,000 (i.e. the approximate increasing is about 12.7 times). English collection will also increase but at a slower rate. Hence, unless the IDF factor is adjusted, documents in small sub-collections are preferred by adding advantageous weights to document frequencies of these documents.

2.3.2 Distributed Architecture

An alternative approach for the MLIR task is the distributed architecture. Two approaches in a distributed architecture can be utilized – according to the type of the retrieval and whether it is bilingual or multilingual, in which all documents are essentially in several monolingual languages (Chen and Gey,

2004b). In the first type, which is used in the multilingual retrieval task, documents in different languages (multilingual document collection) are separated to create several language-specific indices (several sub-collections) in all the target languages. Source queries are then translated to all target languages presented in the multilingual collection. Next, each query (source or translated) is used to perform a monolingual retrieval in its corresponding sub-collection, resulting in individual language-specific ranked lists, which are considered as intermediate results. This step is usually followed by a merging technique so as to merge all individual ranked lists of language-specific documents into a combined single multilingual ranked list, regardless of documents' languages, which in turn will be presented to users. This approach was taken in several studies (McNamee and Mayfield, 2002b; Savoy, 2002; Lin and Chen, 2003). But, merging in bilingual retrieval differs from merging in multilingual retrieval in that the former usually uses a single index, whereas the latter utilizes several indices. Accordingly, the second version of distributed architecture, which is used in bilingual merging rather than multilingual merging, employed putting all documents, regardless of their languages, into a single unified index – as in the centralized architecture. Source queries are translated to all target languages in this single multilingual collection. Next, for a certain source query, each corresponding query in the set of the translated queries – including the source query – is used to search in the huge multilingual index (Chen and Gey, 2004b). Several individual and intermediate lists are returned, as in the first type, and for a given query, results are merged to create a single ranked list using a merging method. However, in such an approach individual documents lists may overlap with each other as a consequence of obtaining all results' lists from a single index.

Regardless of the used approach, the most vital challenge in distributed architectures is the merging of the intermediate results. This is because documents' scores across different language-specific sub-collections are incomparable as each sub-collection uses its own statistics in scoring these documents. Therefore, several merging methods were proposed for traditional distributed MLIR. Among them, the followings are discussed.

1) *Round Robin merging*: In the round robin approach (Voorhees, et al., 1995), the final result list is obtained by taking an item (document) from each intermediate list in turn and in a round robin manner, starting from the top. Similar ranking approaches and an approximately similar distribution of relevant documents and their numbers within individual retrieved lists are assumed.

2) *Raw score merging*: The underlying assumption behind this merging approach is that document scores are comparable across sub-collections in terms of used IR methods and sub-collection statistics (Hiemstra, et al., 2001). Individual result lists are merged and re-sorted according to the raw similarity scores of documents. However, one shortcoming of this merging method is that relevance scores across lists may be incomparable.

3) *Normalized score merging*: Since scores of documents across sub-collections are not comparable, it would be reasonable to normalize them in each individual list before merging. One approach is to divide the score value of each document within each individual list by the maximum score in that list (the score of the first ranked document) (Powell, et. al, 2000; Savoy, 2002). Hence, the approach adjusts documents' scores across lists to values between 0 and 1. Next, all adjusted scores are placed together

and re-sorted according to their new scores. An alternative experimented technique is to adjust the score according to:

$$normalized\_score = \frac{original\ \_score\ - min\ \_score}{max\ \_score\ - min\ \_score} \quad (2.38)$$

Where *min-score* and *max-score* are the minimum and the maximum scores achieved in corresponding language-specific collection. Lin and Chen (2003) proposed also to normalize raw scores by the top-k documents, namely normalized-by-top-k merging method, and at the same time the method incorporates weights based on the degree of translation ambiguity when each source query is being translated based on the premise that accurate translation would likely result in more relevant documents. Translation ambiguity in the approach was determined by the average number of translation equivalents of query terms and the number of OOV words.

4) *Weighted score merging*: Another experimented approach is to adjust documents' scores by employing both their scores and some weighted scores, derived from corresponding sub-collection statistics of documents. This approach is called the Collection Retrieval Inference Network (CORI) for result merging and it was originally developed by Callan, et al. (1995) in distributed information retrieval. The assumption here is that since collections are different, computed scores for them might be used in weight computation. Several solutions based on this method were proposed. One such approach was proposed by Rasolofo, et al. (2001), who developed an approach which assigns each collection a score computed according to the proportion of the length (number of documents retrieved) of result list returned by each collection with a basic assumption that if more documents are found in a certain collection, then that collection would likely contains more relevant documents. In that perspective, Rasolofo and his team developed a merging method that is able to adjust document scores in a way that scores of documents from collections whose scores are less than the average collection score would be decreased, whereas the scores of documents from collections with scores greater than the average collection score would increase. This is done by computing the product of the original document score multiplied by its corresponding collection/language score (weight) as follows:

$$normalized\_score = original\_score * [\ 1 + \frac{S_i - avg\_S}{avg\_S}\ ] \quad (2.39)$$

Where S_i is the ^{i}th collection/language score and avg_S is the mean collection score. Results showed that the performance effectiveness for the same retrieval model was competitive to other DIR merging methods.

5) *Logistic Regression merging technique*: Another method normalizes document scores according to a logistic regression model (Savoy, 2003). A general merging strategy using the logistic regression approach is to predict the probability of relevance of documents according to both the original document score and the logarithm of its rank. The parameters/coefficients for these two values are computed by using a training set and hence fitting a logistic regression model. Finally a single unified list is obtained using these estimated probabilities, which usually are computed independently for each language/collection.

Several studies attempt to compare retrieval effectiveness of the different techniques of result merging. Chen and Gey (2004b) evaluated raw score, normalized score and round robin merging methods. Results showed that the raw-score is the best one among them, whereas both normalized score and raw score methods outperformed round robin merging method. However, as stated by the same researchers, such a conclusion was valid under certain conditions, for example, translated query have similar approximate lengths, even for the source query. Savoy (2002) compared these three merging methods also with the CORI and concluded that normalized score is the leading one. Experiments were conducted using a CLEF test collection.

Distributed architecture is a powerful tool as a monolingual language on both documents and queries is utilized and, thus, better performance is expected. However, the robust feature in this architecture comes from the distribution. Since multilingual collections are distributed in diverse resources in real world applications, it is expected that different IR techniques are adopted with considerable overlapped-indexed portions (Paltoglou, et al., 2008). Furthermore, a large portion of the Web is not indexed by current search engines, known as the invisible Web (Garcia-Monlina and Raghavan, 2001). This is especially true with the explosive growth of the Web. In this invisible Web, owners provides their own search facilities and, thus, search engines are forced to employ these searching capabilities and to combine their results with their own local search engines. This makes the distributed architecture a highly demanded tool. In distributed information retrieval and result merging problem is vital to its performance effectiveness. DIR obtained a lot of attention in recent years but this is beyond the scope of this research.

Comparing centralized architecture to the traditional distributed one, Rasolofo, et al. (2001) showed that it is hard for distributed approaches to obtain the same performance level of when using a single centralized index, stated in Nie and Jin (2002). This is because the retrieval process in the latter would likely perform better because no merging technique is needed. Nevertheless, Chen (2002) showed that using distributed architecture is more effective than the use of a centralized one. However, results varied in the study when using Chinese queries instead of English queries.

2.4 Text Evaluation/Reference Corpora

One of the essential purposes for the use of corpora in IR is the task of measuring effectiveness of ad-hoc IR. The task is often performed by using an evaluation corpus (or several corpora) consists of three types of sets: a set of documents (document collection), a set of topics (queries) and a set of relevance judgments for each query in the query set. In such a case, the corpus is known as a *test collection* (Croft et al., 2010). Among these types of sets, relevance judgments are the most critical parts.

Documents in the majority of the standard test collections have multiple different fields (i.e. title, paragraph, etc) with each document having a unique identification number. Documents are usually represented by some form of markup. Topics present user information needs. In most standard forums, e.g., TREC, search topics in the ad-hoc track are structured in many fields. The three major ones are

(Croft, et al., 2010): *title*, *description* and *narrative*. The title field is a brief query consisting only of few words. Such field is exemplified by the type of searches in Web applications. The description field is a longer sentence(s) of the query and usually contains more informative details about the query. The narrative field is the longest portion of the topic file. It specifies in detail the criteria of judging documents relevance. The narrative field is used by the assessors. Queries are often formulated from the topic files. The relevance of a document with respect to a query specifies the value of that document to fulfill the information need from the user's subjective perspective. To create a test collection, relevance of documents with regards to each topic in the collection should be determined. Relevance judgment usually requires a considerable manual effort and have many aspects, as will be illustrated in the next sections.

This section firstly reviews types of corpora / test collections with examples. This is important for this thesis as a new test collection was created. Since the thesis focuses on bilingual Arabic-English, most examples are also focused on these languages. Next, relevance judgment is discussed in more detail. Following this, evaluation and effectiveness of retrieval measures are presented.

2.4.1 Types of Corpora / Test Collections

Several text collections have been developed to serve as standard test collections and/or reference corpora. The first pioneering experiment for IR evaluation was the Cranfield tests (Cleverdon, 1962), which were conducted over a test collection, known as the Cranfield test collection, contains approximately 1400 abstracts collected from articles of an aerodynamics journal. From that time, many other test collections/corpora, which are different in their sizes, languages, vocabularies and applications, have been developed. Sampled texts in these corpora / test collections are also different from one to another, but usually contain several categories like news, legal articles, hobbies and skills, economies, scientific and specialized texts, reports and religious documents. According to this diversity in their features, corpora / test collections have been categorized into several types (McEnery, et al., 2006), each of which depends on a certain argument. Among these arguments, classification of corpora using their languages (single language vs. multilingual), genres (general vs. specialized) and vocabularies (synchronic vs. diachronic) is discussed.

2.4.1.1 Single Language versus Multilingual Corpora / Test Collections

In terms of their languages, current corpora / test collections can be categorized into two types: single language corpora / test collections and multilingual corpora / test collections (Lin and Chen, 2003). In the single language corpora / test collections, all documents are written in a single language (monolingual). An example of a monolingual collection is the Arabic Agence France Presse (AFP) (Graff and Walker, 2001), which is an Arabic newswire collection acquired from articles taken from the AFP Arabic newswire and created by the Linguistic Data Consortium (LDC). The LDC also released other monolingual Gigaword test collections in different languages, e.g., the fifth edition of the English

Gigaword (Parker, et al., 2011a). Most of them are monolingual and collected from several newswire sources. Examples of monolingual test collections also include some editions of the TREC collections [8], e.g., TREC-3 and TREC-4, which are Spanish monolingual collections collected also from newswire sources. TREC is organized by the NIST.

The second type of corpora / test collections in terms of their languages is the multilingual corpora / test collections. In multilingual corpora / test collections, documents are usually written in several monolingual languages or consist of several monolingual corpora. Such types of multilingual corpora highlight language-specific, typological or cultural features (McEnery, et al., 2006) and they are mostly collected from both newspapers and newswire sources. Parallel and comparable corpora can be also considered as multilingual corpora.

Multilingual test collections are the most dominant in the standard collections and they have become popular after the increasing interest in CLIR since the latter has a major impact on test collections and corpora. The most widely known of multilingual test collections are the different editions of TREC. It contains several monolingual test collections in different languages along with their queries and relevance judgments. Arabic has been included in TREC in 2001 in the same cross-lingual track (Gey and Oard, 2002). TREC-2001 was collected from Arabic newswire sources taken from the AFP.

NII Test Collection for IR Systems (NTCIR)[9] contains languages of the East Asian region (i.e. Chinese, Japanese and Korean) and their collections are of similar sizes to TREC. NTCIR focuses on CLIR. Many of the NTCIR collections are acquired from newspapers and news articles such as NTCIR-3, NTCIR-4, NTCIR-5 and NTCIR-6. Queries in this collection are in Chinese, Korean, Japanese and English. It is observed that some NTCIR collections include many scientific documents, although they may be placed with newswire documents. For example, the document collection in the NTCIR-1 consists of abstracts of the proceedings of academic conference papers from 1988 to 1997; more than half are English-Japanese paired documents. Furthermore, some documents in the NTCIR, mostly non-English ones, are mixed with bilingual keywords or paired snippets of texts. Most of these multilingual test collections, even those containing multilingual and mixed documents, are built to help with retrieval of documents based on monolingual queries, even if they are translated, as in the CLIR track. Thus, their query sets are monolingual or/and the multilingualism characteristic in their multilingual documents are handled as if these documents are monolingual.

2.4.1.2 General vs. Specialized Corpora/ Test collections

In terms of vocabulary types, corpora/ test collections can be classified as general corpora/ test collections or specialized corpora / test collections. A general corpus/ test collection, as the name indicates, usually contains different genres and domains such as regional and national newspapers, legal documents, encyclopedias and periodicals. In addition, general corpora / test collections may contain

[8] http://trec.nist.gov/

[9] http://research.nii.ac.jp/ntcir/index-en.htm

written or spoken data. TREC, NTCIR and CLEF (European Cross Language Evaluation Forum)[10], which is another valuable series of test collections and focuses on European languages and CLIR, can be considered to be general test collections because test documents in them are mostly of general domain news stories (Rogati and Yang, 2004). In contrast, a specialized corpus/ test collection contains terminology in a specific domain. However, this specialization does not have defined boundaries. Instead it should contain particular types of texts. Examples of specialized test collections include CACM (Dunlop and Rijsbergen, 1993), which was built from titles and abstracts of the Communications of the ACM from 1958-1979 along with its query set and relevance judgments. Hmeidi, et al. (1998) built an Arabic test collection with 242 abstracts gathered from the proceedings of the Saudi Arabian national computer science conference. The document collection is very small and contains only titles and abstracts and collected from a certain country, rather than from a region. As illustrated in the previous sub-section, NTCIR also contains some specialized documents, such as in NTCIR-1 and NTCIR-2, which primarily consist of abstracts of academic conference papers. However, NTCIR changed its consequent editions of the released collections, particularly after NTCIR-2, to newspaper/newswire sources (Gey, et al., 2005). This was justified because such orientation is a normal reaction to social requirements such as the growing interest and emergent importance of technological information in business sectors. Patent collections were created for the patent test in IR. Most such patent collections consist of several monolingual and/or parallel unexamined patent texts or abstracts in several monolingual and/or parallel languages, although many patents are multilingual.

The HKUST (Hong Kong University of Science and Technology) is another English specialized corpus containing exam papers and essays collected from texts written by Chinese students of English at the computer department of the Hong Kong University of Science and Technology (Milton and Tong, 1991). The HKUST is not available for distribution or downloading and it was created to reveal the development of English teaching materials, rather than for the IR task.

Springer is another specialized and parallel test collection of English-German in the medical domain[11]. It contains 9640 documents, with 1 million tokens in each language, constructed from titles and abstracts of medical journal articles in English and in German, along with their queries and relevance judgments.

It is noted that the majority of scientific corpora / test collections are monolingual and the Arabic language is rare among them.

2.4.1.3 Synchronic vs. Diachronic Corpora / Test collections

Synchronic corpora / test collections are often used to compare regional varieties, whereas diachronic, or historical, corpora are usually used to compare vocabulary from the same language gathered from a wide area and different time periods (McEnery et. al, 2006). To study regional variation in monolingual Arabic documents, Abdelali, et al. (2005) constructed a large synchronic test collection in Modern Standard

[10] http://www.clef-campaign.org/

[11] http://muchmore.dfki.de/resources_index.htm

Arabic (MSA), which is a modern version of the Arabic language that is usually used in formal communications, from different regional Arabic newspapers.

The ICE (International Corpus of English)[12] is another synchronic and monolingual corpus, which is gathered from both written and spoken English after 1989 in different countries (United Kingdom, South Africa, United States of America, Canada, India, Philippines, Hong Kong, etc). Each team of the ICE in a given country creates a sub-corpus of one million words and, thus, the entire corpus contains various regional variations in modern English.

The Bibliotheca Alexandria (BA) Library in Egypt initiated an ambitious project to build the Arabic version, namely the International Corpus of Arabic (ICA) project [13], of the ICE on the same principles. The corpus is intended to be representative of MSA across the Arabic world with a primary goal to support research in the Arabic language. According to the planned design (Alansary, et al., 2007, 2008), the targeted size of the ICA is 100 million words and the planned sources and genres, from which the corpus would be collected, contain, for examples, newspapers, magazines, novels, net articles and electronic press. However, the final compilation of the corpus would contain a genre category and sub-category, e.g., humanities and history. Although neither the current size of the ICA nor its distribution are known yet, Alansary, et al. (2008) analyzed a sample of the initial version and showed a road map and some technical information on issues like how the corpus will be analyzed in terms of morphological and semantic analysis, pre-analysis and full text analysis, for example, which stem-based approach will be used, as well as the selection and the description of the models that would be used in these analysis processes. The ICA project is still in progress.

2.4.2 Relevance Judgment

Relevance judgment is complicated because it is essentially subjective, as it can vary according to the person who makes the assessment or for the same person at different times (Voorhees and Harman, 2001). It might be established according to how up-to-date the document is or document's availability or its subject or even according the degree of relevance with which document matches the information need. Despite these distinctive complexities, analysis of many well-known forums, e.g., TREC has revealed that relevance judgments are complete enough to conclude about the relative performance of IR systems. In other words, diversities in relevance judgments do not have a major effect on the error rate for comparisons (Croft et al., 2010). Furthermore, it is assumed that relevance is determined by the topicality of documents.

Whenever relevance judgments are created, it is important to know total number of relevant documents for each topic. But, this is infeasible, especially for large test collections, due to the large effort needed. As a result, a sample of documents for each topic is only assessed. This is known as pooling (Spärck-Jones and Rijsbergen, 1975). In the pooling technique, the top *k* documents, e.g., 100 retrieved by each

[12] http://www.ucl.ac.uk/english-usage/projects/ice.htm

[13] http://www.bibalex.org/unl/frontend/projects.aspx?id=9

participating retrieval algorithm are collected and all these selected documents are pooled together into a single pool. Documents that were not selected in the unified pool are often considered as irrelevant. Duplicates in the pool are removed and documents are presented to assessors in a random order without any information about which document was returned by which algorithm or what rank a document obtains. Although, the pooling method has been questioned since documents not in the pool are handled as irrelevant, even if they are relevant, the analysis of (Buckley and Voorhees, 2000) showed that the technique is stable and sufficient to acquire accurate comparisons and it is useful in measuring effectiveness of IR systems. In his study to explore pool quality and the potential bias of using such technique, Zobel (1998) concluded similar trends in that the TREC collections were not biased against unjudged runs.

The scale of the employed relevance judgment is another issue of complex nature of documents' relevance. Generally, a binary level of relevance judgment (relevant or irrelevant) is utilized with a simple criterion for accepting a document as relevant, that is a document is judged as relevant if it contains any type of information that could be used in writing a report on the subject of its corresponding topic. This is often done without paying account to the number of other documents that contain the same information (Voorhees and Harman, 2001).

The majority of relevance judgments in TREC collections are of binary scale (Croft et al., 2010). In such a case, the retrieval task concentrates on higher recall, where it is important to retrieve any relevant document. However, for some tasks, like to what degree a document is relevant to the query, multiple levels of relevance (graded non-binary relevance) can be used. For example, relevance assessment for a particular document with respect to a specific query can be done on a four-point scale (0-3), with 0 = irrelevant, 1=marginally relevant, 2= good and 3=excellent. In such non-binary relevance, the retrieval task emphasizes highly relevant documents or document d_i should be ranked higher than document d_k because it is more relevant. Cited in Kekäläinen (2005), Tang, et. al. (1999) stated that dividing relevance scale into suitable numbers of degrees has been explored and it is concluded that there is no clear answer for such numbers of degrees as this solely depends on the required levels of accuracy and the type of the desired retrieval task as well.

In judging relevance, it is important also to consider the number topics to use. The Analysis of TREC experiments (Buckley and Voorhees, 2000) concluded that using 25 queries would result in invalid conclusions when comparing the effectiveness of two algorithms, but the use of 25 queries is a minimum. However, as a rule of thumb, the use of 50 queries is sufficient (Manning, et al., 2008). If a difference of 0.05 in the Mean Average Precision (MAP), which is illustrated in the next section, between two IR techniques occurs and 50 queries are used in their experiments, then the conclusion that the performance of one technique is better the performance of the other, would be of an error rate below 4% in comparisons (Croft et al., 2010). Thus, it is concluded that using 50 queries is sufficient to ensure a low error rate.

2.4.3 Evaluation Measures

The performance of an IR system can be measured in different ways, depending on retrieval task and used relevance judgment (Croft et al., 2010). If the binary relevance judgments are employed for assessing documents, then *precision* and *recall* measures can be used. The precision is the ratio of the number of retrieved relevant documents over the total number of documents retrieved. More formally:

$$Precision = \frac{\text{number of retrieved relevant documents}}{\text{number of retrived documents}} \tag{2.40}$$

The recall is defined as the fraction of relevant documents that are retrieved. Formally:

$$Recall = \frac{\text{number of retrieved relevant documents}}{\text{number of relevant documents in the collection}} \tag{2.41}$$

For a certain query, the precision evaluates the ability of the IR system algorithm to eliminate non-relevant documents, whereas the recall evaluates its ability on retrieving all the relevant documents. When the precision increases the recall typically goes down and vice versa. The two measures assume that users would like to retrieve relevant documents as much as possible, while irrelevant documents should be minimized as much as possible. This is especially true for systems that often retrieve a fixed set of documents and often make binary decisions, without considering relevance ranking – as in Boolean models. This makes the computations of precision and recall simple, however, it does not differentiate between the ranking (Croft et al., 2010). For instance, if there are two relevant documents at rank 1 and 3 in a particular algorithm and the same documents are ranked at 9 and 10 in another algorithm, then the two algorithms would have the same precision and recall values at rank 10. Accordingly, precision and recall values are usually computed at a predefined rank position, e.g., at rank p), instead of computing them, meaning precision and recall, at every rank position. But, for any two algorithms if the value of the precision at rank p of one algorithm between them is higher than the value obtained from using the same measure with the other algorithm, then the recall of the first algorithm would be higher also the peer recall of the second algorithm. Therefore, only precision at a predefined rank position, e.g., 10 and 20 is often used (Manning, et al., 2008; Croft et al., 2010). This is called *precision at p* measure. Note that using this measure changes the search task to focus on retrieving the most relevant documents at a given rank, rather than finding all the relevant documents (Manning, et al., 2008; Croft et al., 2010). But, the measure also may not differentiate differences in the ranking at positions 1 to p. Therefore, precision is often used at 11 standard recall levels: 0, 0.1, 0.2,…, 1.0. Some interpolation mechanism is used also, in order to obtain the precision values at all the standard recall levels. If the values of the precisions from the rank positions are averaged for each query, where a relevant document is retrieved, then the measure is called the *average precision*, which is a single summarization value. The overall conclusion about the performance of a specific algorithm is then determined by averaging the values of the average precision over a sufficient number of queries (Manning, et al., 2008; Nie, 2010). The used

measure is known as the Mean *Average Precision* (MAP). The MAP is the most widely used measure in evaluation of IR and CLIR systems (Croft et al., 2010; Nie, 2010).

When graded relevance is used, the *Discounted Cumulative Gain* (DCG) can be used (Jävelin and Kekäläinen, 2002). The DCG is becoming an increasingly popular measure for evaluating performance (Croft, et al., 2010; Nie, 2010). The assumption in this measure is that lower ranked documents (documents with greater ranks) are less valuable for users and less likely to be tested by them. In that perspective, the most relevant documents (highly relevant) are more valuable than those documents with marginal relevance. Thus, if a graded relevance scale is used to judge the relevance of documents, then it can be employed by the DCG as a measure the value level or gain from testing a document. Thus, from the top of the list the gain begins to accumulate and it may be reduced or discounted as other documents are examined.

Thus, the DCG is the total gain accumulated at a particular rank p (Jävelin and Kekäläinen, 2002; Croft, et al., 2010). Thus, DCG at a particular rank p (DCG_p) is defined as follows:

$$DCG_p = R_1 + \sum_{i=2}^{p} \frac{R_i}{\log_2 i} \quad (2.42)$$

Where R_i is the graded relevance level of the document retrieved at rank i. The denominator $\log_2 i$ is the *discount* of the gain. Since the focus of this measure is on the top ranks, the TREC standards stated that the values of p are typically small, such as 5 or 10. To conclude the performance effectiveness of a certain algorithm, the DCG values are averaged across the employed set of the search queries. If the DCG value at each rank for a certain query is divided by the DCG value that is obtained from the perfect ranking of documents using the same query (i.e. listing relevance level starting from the highly relevant one such as 4,4,4,3,3,2..), this would result in the *Normalized Discounted Cumulative Gain* (NDCG) measure (Jävelin and Kekäläinen, 2002). In his study for evaluation measures sensitivity, Sakai (2007) concluded that NDCG is the best evaluation measure for document rankings. The experiments were conducted using NTCIR collections.

Beside theses measures, it is also common to measure effectiveness of retrieval algorithms by comparing their performance to that obtained from conducting a monolingual retrieval run that makes use of manually translated queries to target language, by human experts. Such an approach would result in a strong run that can be considered as an upper baseline, which is in most cases unreachable by the algorithms under evaluation. However, such an evaluation is often assessed in terms of percentages computed for the performance of both the proposed technique and the monolingual retrieval.

2.4.4 Significance Test of Retrieval Performance

In the context of IR, it is important to know if there is an improvement of a particular retrieval algorithm over another and whether this improvement is caused by a real difference between the two algorithms or the difference has just appeared by chance. Such difference between algorithms is often measured using statistical significance tests. In particular, in IR the concern is in the paired tests, as the algorithms under evaluation use the same set of queries, which in turn should be of a reasonable amount.

When a statistical test is used for comparing performance of two ranking algorithms (say algorithm *X* and algorithm *Y*), a typical confidence level of a 95% is utilized. This value means that in 95% of choices of *X* and *Y* the performance of *X* will go above that of *Y*. In other words, if the probability of the observed difference between the algorithm *X* and the algorithm *Y*, known as significance value, is small enough (i.e. < 0.05), then this difference is considered as significant as there is 5% probability of being false positive. Since the significance value represents the probability of error in accepting that the result is correct, the value 0.05 is considered as an acceptable error level.

The most commonly used significance tests in IR are the Student's t-test and the Wilcoxon signed-rank test (Croft et al., 2010). However, in spite of the fact that the t-test assumes a parametric distribution, many studies, e.g., (Sanderson and Zobel, 2005), showed that it could correctly distinguishes between rankings of two algorithms.

2.5 Other Related Work: Bilingual Querying

The issue of using bilingual queries and documents has been discussed in library science but, in terms of exploring Web search behavior using different techniques, e.g., interviews and query analysis. Hansen, et al. (2002) enumerated some user requirements for Cross Language Information Retrieval (CLIR) systems, including the support of multilingual queries and the ability to search multiple languages simultaneously. Petrelli, et al. (2004) found that the English term in a multilingual query is usually utilized as a pivot in searching because English is still the dominant language in technical jargon. Furthermore, for searching, users often choose the language that suits their needs and that they are familiar with. Such language is not always the native language of those users.

Rieh and Rieh (2005), in their study of Web searching behaviour, concluded that the querying and searching behaviour is dependent on users' needs, purposes of searching and users' ability to speak a foreign language. Thus, the searching behavior of science scholars is different from that of scholars in humanities and social sciences, who often use their native languages. Some users may post queries in their native languages or a foreign language while others prefer to enter multilingual queries. Furthermore, users may also prefer to search for their information need in different languages separately, as they are not confident with translation accuracy. Rieh and Rieh concluded that it becomes important for research to expand the understanding of multilingual Web searching from the user side.

In his study to analyze Web users' behaviours, Lu, et al. (2006) tackled the reasons behind using multilingual trends of querying in users' behaviour. The findings, which were extracted from the analysis of a query log of a search engine and more than 77,000 multilingual queries, showed that mixed query searching between Chinese and English was primarily caused by the following: using computer technologies, names of magazines and firms; some Chinese words do not have a popular translation; and the culture, such as in Hong Kong, of using both Chinese and English in speaking and writing. Analysis by Lu and his team also showed that there were many queries that consist of both a Chinese term and its corresponding translation in English. Users in such cases might intend to get a higher recall.

Fung, et al. (1999) observed that the colloquial Hong Kong language is mixed between Cantonese and English words. Therefore, based on the fact that mixed queries often consist of a primary language and secondary language, Fung and his colleagues proposed a mixed-language query disambiguation technique. The technique utilized co-occurrence information of words between those in the primary language and those in secondary language. In particular, the word in the secondary language in the mixed query is translated to the primary language and then a co-occurrence information is extracted for each translation of a secondary language in the query with the words in the primary language in the same mixed query. The co-occurrence information was computed using a monolingual document collection and a bilingual dictionary. In the study, different approaches were tested, including the use of a window sentence, instead of employing only neighbouring words. Results showed that the proposed approach was better than a baseline that utilized only neighbouring words for disambiguation. The same approach was also followed by Cheung and Fung (2005) and the same conclusions were also reported.

2.6 Summary

This chapter reviewed major issues in CLIR and discussed various techniques that are adopted in current research. It is observed that the query translation approach is the common focus for the majority of studies in CLIR. A number of translation resources and techniques, however, can be used for this purpose, each of which has its strengths and weaknesses. The dictionary-based approach for query translation with token-to-token mapping is the most widely used method. This is because MR dictionaries are abundant, readily-available in many several languages, easier to acquire and provide us with enough recall. Moreover, bilingual dictionaries neither require us to form syntactic sentences nor need a sufficient training data with a high quality system for translation, as in the MT approaches. Bilingual dictionaries also are inexpensive, unlike parallel and comparable corpora, and they, almost, do not need license keys - as in the search engines libraries to access results' lists. However, in the MRD realm, and also in some other translation resources, the important research difficulty is that a term in a source language often has more than one translation, without any contextual information and, hence, results in translation ambiguity. Several approaches had been proposed for this problem. Some approaches attempt explicitly to select the best translation, according to some observed phenomenon ,e.g. co-occurrence statistics that are derived from parallel corpora / document collections or from unlinked and monolingual corpus in the target language, while others attempt either to re-weight query terms (in terms of TF and DF), base their matched translations on a dictionary for example, when their no translation probability knowledge is available, or they attempt to estimate a probability for each possible translation candidate (when such probabilistic knowledge is available) and, hence, the translation took place implicitly during retrieval, e.g., different variants of structured query model such probabilistic SQM. Both of the approaches have shown to be valuable and significantly effective for CLIR task. Approaches that make use of the inverse document frequency of terms to distinguish between specific (with few number of translations) and general terms have also shown to be effective especially when they are merged with some statistical models.

Approaches to traditional MLIR were also reviewed in this chapter. It was shown that the centralized approach is effective as documents are placed into a single index, while the traditional distributed architecture usually needs a merging method to provide the final single list. The merging problem makes it not easy for the latter approach to reach the same effectiveness of the centralized approach, which also has major drawback of overweighting. The state-of-the-art test collections were also reviewed in this chapter. Several test collections were developed for many languages but, it was observed that the majority of them were created from the news genre domain and they are almost monolingual with monolingual query sets. Furthermore, scientific test collections among these test collections are rare and Arabic is not one of them. The evaluation measures for CLIR systems were also discussed in this chapter and it was shown that there are some measures becoming increasingly popular, such as the DCG. Bilingual mixed queries were also presented in the chapter. It was shown that the area of multilingual querying is relatively ignored in current research, as the majority of the techniques shrink the CLIR task, and also the MLIR one, to a monolingual retrieval preceded by a translation, meaning that the query is usually issued in a monolingual form. Accordingly**,** most CLIR and MLIR are based on monolingual weighting, making them either language-unaware systems or not adequate to handle mixed queries.

The next chapter describes current approaches in Arabic information retrieval.

CHAPTER

3

Arabic Information Retrieval: State-of-the-art

Over the last two decades Arabic IR, either monolingual or cross-lingual, has become one of the popular areas of research in IR (Moukdad, 2006; Abdelali, 2006), especially with the explosive growth of the language on the Web and the emergence of the CLIR field, which shows the need to retrieve documents in other languages. This increasing interest in Arabic, however, is caused by its morphology, which is radically different from the European and the East Asian languages (Xu, et al., 2002; Moukdad, 2006; Larkey, et al., 2007; Salhi and Yahya, 2011). It is also caused by the geographical and economic impact of the Arabic regions, in which Arabic is the official spoken language (Abdelali, 2006).

However, in spite of the significant achievements and developments in existing Arabic text retrieval systems, its support is comparatively poor and much weaker than for English (Abdelali, 2006, Alansary, et al., 2007, 2008; Salhi and Yahya, 2011). In particular, Arabic is still lacking in high quality IR and NLP tools, e.g., the need for efficient machine translation systems. Nevertheless, there has been some important progress in Arabic IR. Many stemming techniques, POS taggers and other algorithms have been proposed for this language.

This chapter reviews the state-of-the-art in solutions and techniques that are used for Arabic IR within its two categories: monolingual and cross–lingual. It presents the current solutions and approaches that are proposed for some major challenges in Arabic IR, along with a considerable number of studies that have implemented these approaches. In addition, the chapter gives a strong base for the question of why Arabic causes a significant challenge for text IR systems and it sheds some light on the most essential Arabic rules that affect Arabic retrieval and how these rules can lead to ambiguity.

The remainder of this chapter is organised as follows. Section 3.1 describes the basic characteristics of the Arabic language and its morphology. Following this, section 3.2 discusses the challenges that hinder IR tasks in Arabic and the different levels of ambiguities that arise because of Arabic morphology.

Morphology is pinpointed for the purpose of understanding how to develop effective Arabic IR systems. Section 3.3 is dedicated to illustrating the current solutions that have been employed for Arabic monolingual information retrieval, such as stemming and tokenisation. In the same section, other complementary techniques that are used to improve Arabic IR are described. Section 3.4 describes some of the complementary techniques that are used to enhance Arabic IR. In particular, the section discusses regional variation and broken plural problems. The latter problem is related to Arabic morphology, as will be shown later. Section 3.5 illustrates some of the utilized techniques in Arabic cross-lingual information retrieval. Approaches to translation and transliteration are illustrated in this section. Finally, in section 3.6 the chapter is concluded.

3.1 The Arabic Language

Arabic is one of the oldest languages that originated in the Arabian peninsula in pre-Islamic times. It belongs to the class of Semitic languages, which also includes Hebrew, Aramaic and Amharic and its first documented inscription was found around 328 C.E (Arabic History, 2012). There is an evidence that the Arabic script was derived from the ancient Nabatean (Aramaic) alphabet, but the language flourished independently with the rise of Islam in 622 (Arabic History, 2012) as the language became, at that time, the lingua-franca of a large group of people, instead of its use by a few tribes in the Arabian peninsula.

Letters in Arabic were originally written without dots. Dots were added to the language for disambiguation purposes since the seventh century, with the emergence of Islam, when more non-Arabic speakers begin to speak the language. Later, in the same century, diacritical marks (short vowels) were invented and added to minimize ambiguity.

In contemporary time, Arabic can be classified into three forms (Saad and Ashour, 2010): Classical Arabic (CA), Modern Standard Arabic (MSA) and Dialectal Arabic. Classical Arabic was the language of old Arabic-speaking people, e.g., pre-Islamic times and during the appearance and rise of Islam. A typical example for classical Arabic is exemplified in the Holy Quran. Modern Standard Arabic, known also as *Fusha*, is a modified version - with a modern vocabulary - of Classical Arabic. It is typically found in news papers. MSA is also used in official speech and communication in the Arabic region. Additionally, it is the formal language of the media and education across the Arabic world. Dialectal Arabic, as the name indicates, is used in informal communication in all Arabic-speaking countries and its vocabulary is regionally variant. Accordingly, the term 'Arabic' refers to both MSA and Dialectical Arabic (Abdelali, 2006).

Arabic is the official language in the Arabic region, which includes 22 countries as mentioned above (Mirkin, 2010). It is estimated that there are three hundred and fifty nine million first-language speakers of Arabic (Mirkin, 2010). Since it is the language of religious instruction in Islam, many other speakers from varied nations have at least a passive knowledge of the language. Arabic also is one of the six official languages of the (UN) and it is the fifth most widely used language in the world (Chung, 2008). There is an interest in Arabic as the Arabic world is one of the most influential regions on the world. It is

commonly known that this is mainly due to economic impact, for example the first world reserve of oil and gas, and geographical reasons, e.g., the Strait of Bab Elmandab in Yemen and Suez Canal in Egypt.
Sentences in Arabic are delimited by periods, dashes and commas, while words are separated by white spaces and other punctuation marks that are mostly similar to those in English, e.g. comma, hyphen and question mark. Arabic script is written from right-to-left while Arabic numbers are written and read from left-to-right. For example, 2013 in Arabic is read and written as in English starting from left to right. Script of Arabic consists of two types of symbols (Habash and Rambow, 2007): these are the letters and the diacritics (known also as short vowels), which are certain orthographic symbols that are usually added to disambiguate Arabic words. For instances, SEEN (س) is a letter equivalent to 'S' in English, whereas سُ is a diacritized letter with the sound 'su', like in the word Sudan. Short vowels are always omitted in written MSA texts as Arabic speakers could distinguish easily between words with similar forms from the context in which they occur.
Basically, the Arabic alphabet has 28 letters (Tayli and Al-Salamah, 1990) and, unlike English, there is no lower and upper case for letters in Arabic. An additional character, which is the HAMZA (ء), has been also added, but, usually it is not classified as the 29th letter. Table 3.1 illustrates the complete set of the Arabic alphabet. Each of the letters in the set can be extended using short vowels, resulting in approximately 90 elements (Tayli and Al-Salamah, 1990). For example, the letter SEEN can have the sound 'sa' (written in Arabic as سَ), 'su' (written in سُ) and 'si' (witten as سِ).

ص	ش	س	ز	ر	ذ	د	خ	ح	ج	ث	ت	ب	أ
ي	و	هـ	ن	م	ل	ك	ق	ف	غ	ع	ظ	ط	ض

TABLE 3.1: The complete set of the Arabic letters.

Letters can have different glyphs depending on their positions within words and their preceding and following letters. For example, Table 3.2 illustrates different writing styles of the letter GEEM (ج). Among the 28 letters, only three are vowels and the remaining ones are consonants. These vowels are the letters: *YAA*, *ALIF* and *WAW* (أ, ي and و), which are also known as long vowels.

Single	First Position	Middle Position	Final Position
ج	جـ	ـجـ	ـج

TABLE 3.2: Illustrates different writing glyphs of the Arabic letter JEEM (ج).

Like all Semitic languages, Arabic has different entirely morphology when compared to Indo-European languages. It is highly derivational and context dependent with complex morphology. The derivational system in Arabic morphology is based on roots. The majority of roots consist of three consonants. Words are formed by expanding the root with affixes using well-known morphological patterns (known sometimes as measures). This is similar to the derivation process in English, although it is more restricted in Arabic. For example, the English words *universe*, *universal*, *universally* and *university* have different meanings but they share the same basic sense. Unlike English, the derivational system, which is covered

later in this chapter, could result in a large number of verbs and nouns. This is because the affixation system of Arabic is rich, especially with the large number of the available affixes. For example, Table 3.3 shows some different forms derived for the word أخلاء, which is the plural of the word خليل (meaning: a close friend) after being attached to different affixes. All words are correct in MSA. This feature causes Arabic to have more words that can occur only once in text, compared to other languages, e.g., English (Goweder and De Roeck, 2001).

Word
أخلاء
أخلائه، أخلاؤه، أخلاءه، أخلائهم، أخلاءهم، أخلاؤهم، أخلائهم، أخلائهن، أخلائهما، أخلاؤهما، أخلاءنا، أخلائنا، أخلاؤنا، أخلائكم، أخلائنك، أخلاءك، أخلاؤها، أخلاؤها، أخلائها،أخلائي، وأخلائي، الأخلاء، بالأخلاء، بأخلاء، بأخلائهم ... إلخ

TABLE 3.3: Different affixes attached to Arabic word أخلاء (meaning: the plural of the word خليل, which means 'a close friend').

Arabic words are classified into three main parts-of-speech: nouns (including adjectives and adverbs), verbs and particles. Particles in Arabic are attached to verbs and nouns. Words in Arabic are either masculine or feminine. The feminine is often formed differently from the masculine, e.g., مُبرمج and مُبرمجة (meaning: single masculine programmer and single feminine programmer, respectively). The same feature appears also in both nouns and verbs in literary Arabic in order to indicate number (singular, dual 'for describing two entities' and plural) as in مُبرمج, مُبرمجان and مُبرمجون (meaning: singular programmer, two programmers and more than two programmers, respectively).

Arabic also has three grammatical cases, as well. These cases are: nominative, accusative and genitive. For example, if the noun is a subject, then it will have the nominative grammatical case; if it is an object, the noun will be in the accusative case; and the noun will be in a genitive case if it is an object for a preposition. These grammatical cases cause Arabic to derive many words from a single noun (i.e. adjective) because it often results in a different form of the word. Note that adjectives in Arabic are nouns as was shown in the previous paragraph. Table 3.4 on the next page illustrates the different forms that can be derived from the adjective مزارع (meaning: farmer). In the table, the symbol (*) is added to words in which only diacritics can be used to distinguish among words. Verbs in Arabic are also formed in a similar derivational process. For example, from the tri-literal root زرع (meaning: farm) many other verbs can be formulated.

Arabic also has a high syntactical flexibility (Abdelali, 2006). For example, the use of different prepositions with the same word while preserving the meaning is common, e.g., in sentences like تخرج في and تخرج من (meaning: graduated from), two different prepositions من and في are used with the same verb تخرج but the meaning is similar. Additionally, In Arabic, plurals can be of different forms, all them are correct. For example, the plural forms of the word جزيره (meaning: island) can be either جزر or جزائر. Furthermore, words in Arabic sentences are of free order, and, thus, they can be swapped in many cases. For example, in the sentence قاد الولد السيارة بسرعة (meaning: the boy drives the car quickly), the order of words are is of the form 'verb-subject-object'. The same sentence, however, can be written as الولد قاد السيارة بسرعة (in the form of subject-verb-object) or قاد السيارة الولد بسرعة (in the form of verb-object -subject).

#	Word	Description
1	مزراع	Singular masculine in nominative, accusative and genitive cases
2	مزارعة	Singular feminine in nominative, accusative and genitive cases
3	مزارعان	Dual masculine in nominative case
4	*مزارعَين	Dual masculine in accusative and genitive cases
5	مزارعتان	Dual feminine in nominative case
6	مزارعتين	Dual feminine in accusative and genitive cases
7	مزارعون	Plural masculine in nominative case
8	*مزارعِين	Plural masculine in accusative and genitive cases
9	مزارعات	Plural feminine in nominative, accusative and genitive cases

TABLE 3.4: Different derivative forms from the adjective مزارع (meaning: farmer).

In addition, Arabic also allows fronting of words, which means that the second part of a sentence, adverbs in English for example, can be written at the first of the sentence. For example, the previous sentence can be written by fronting the word بسرعة, resulting in بسرعة قاد الولد السيارة. The approximate meaning for this sentence when fronting is used would be something like (quickly, the boy drives the car).

Beyond the known challenges of processing natural languages, the Arabic features provided above make the IR task more challenging.

3.2 Arabic Challenges to Information Retrieval

In information retrieval, it is generally accepted that ambiguity in Arabic is greater than in many other languages. Ambiguity in Arabic is caused by one or more of the following five features of Arabic (Tayli and Al-Salamah, 1990).

3.2.1 Orthographic Variations

Variation in Arabic is common and presents a challenge for both Arabic monolingual and cross-lingual information retrieval. Orthographic variations in MSA, and also in colloquial Arabic, has six levels (Zawaydeh and Saadi, 2006). Figure 3.1, adapted from Zawaydeh and Saadi (2006), illustrates these levels of orthographic variations.

Typographical variations are merely caused by the Arabic letters ALIF with its different glyphs (أ, إ, آ and ا) and YAA with its dotted and un-dotted forms (ي and ى) and HAA with the forms ه and ة . In most cases, one of the glyphs of a certain letter is altered/dropped, initially, medially or finally, with another glyph of the same letter when writing text (Buckwalter, 2004).

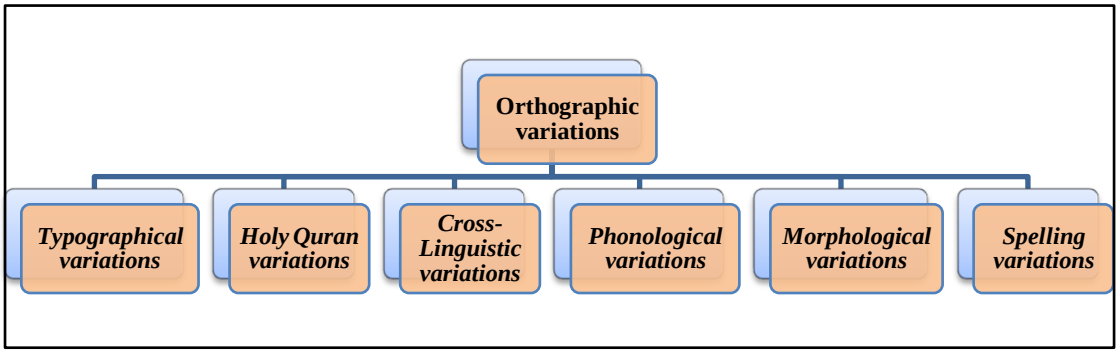


FIG. 3.1: Types of orthographic variations in MSA.

Typographical variations also occur due to altering the letter TAA MARBOUTA with HAA or vice-versa. Table 3.5 shows some examples of different typographical variations in MSA. Sometimes the typographical variant changes the meaning of the original word significantly, for example the قرآن (meaning: the Holy Quran) is typographically changed to قران (meaning: marriage contract), when the letter ALIF MADDA glyph in the middle is changed to bare ALIF.

MSA	Variant	Gloss	Typographical Occurrence
امتحان	إمتحان	exam	The final bare ALIF is changed to ALIF HAMZA below
صفاء	صفا	purity	The final HAMZA is dropped
قرآن	قران	the Quran	ALIF MADDA in the middle is altered to bare ALIF
علاء	علا	a proper noun	They compute (plural feminine)
نافذه	نافذة	window	The final letter HAA is altered to a different letter, which is TAA MARBOOTA
زراعي	زراعى	agricultural	The final dotted YAA is changed to un-dotted YAA

TABLE 3.5: Illustrates some examples for typological variants in Arabic.

Holy Quran variations have been classified as a separate branch because they implement different rules for deletion and substitution of letters. For example, the letter ALIF is deleted when it appears in the masculine and feminine forms of regular plurals, for example المؤمنت (meaning: female believers) instead of المؤمنات. Sometimes the letter NOON (ن) is altered by the letter MEEM (م), e.g., مم بعد instead of من بعد. The Holy Quran variants, however, are only particular to the Quran. They are not contained in MSA.

Cross-Linguistic variations result primarily from transliteration of foreign words in Arabic script. For instance, many technical terms in Arabic are written in different transliterated forms, e.g., the transliterated Arabic words for the English word computer are كمبيوتر and كومبيوتر.

Phonological variations occur due to regional and colloquial variations. Regional and colloquial variations add to the complexity and ambiguity in Arabic information retrieval since Arabic is the mother tongue language in 22 countries. The different dialects, however, vary considerably from one another

and from modern standard Arabic at all levels of language, i.e. pronunciation, phonology, vocabulary, morphology and syntax (Habash and Rambow, 2006).

Morphological variations may occur as a result of unclear boundaries of words in colloquial transliteration, e.g., أقولك (meaning: I say to you), instead of أقول لك. Omission of the final ALIF after the WAW from the 3rd masculine plural suffix also contributes in producing morphological variants. For instance, the word ذهبوا (meaning: they went) may erroneously be written as ذهبو. Another crucial problem in morphological variants that causes significant ambiguity is the problem of free concatenation of words, known also as the run-on words problem (Buckwalter, 2004; Zawaydeh and Saadi, 2006). Run-on words occur when the preceding word ends with a non-connecting letter such as DAL, ALIF, WAW or RAA. For example, sometimes the word ود مدني (a city name) is written as ودمدني without space between the word ود and the word مدني. The most common "run-on" words appear in the combination of the words: لا and ما (both are used for negation) with perfect or imperfect verbs (meaningless verbs). It is very frequent to find imperfect verbs like مازال and لازال instead of ما زال and لا زال with spaces, respectively. In this example, the imperfect verb, known also as copula (Attia, 2008), زال (meaning: being removed) is used with the word ما and the absolute negation لا. Another two cases of run-on words is in proper names, as in عزالدين instead of عز الدين, and in descriptive numbers, as in أحدعشر instead of أحد عشر (meaning: eleven).

Spelling variations, in terms of errors, occurs usually as a result of misspelling of MSA words. For example, the word خطأ (meaning: mistake) may erroneously be written as خطاء. But, differently to other languages, spelling variants may also occur due to the fact that some Arabic letters, whose phonetics are similar, can be written in different forms. For example, the words رضا (meaning: satisfaction) and رضى have the same meanings with the same phonetics but the final letters in both are different, as both the un-dotted YAA (known in this case as ALIF MAKSURA) and the bare ALIF have the same phonetics.

Orthographic variations in Arabic play a big role in confusing IR systems and leads to a larger possibility of a mismatch between queries and documents. This is true for all types of variations. Xu, et al., (2001) stated that ignorance of such kinds of orthography produces ambiguous words and may result in an invalid word that is un-stemmable by some stemmers. According to Beesley (1998), cited in Aljlayl and Frieder (2001), ambiguous written words have an average of five valid morphological analyses per word. Thus many non-relevant documents will be retrieved, whereas many relevant ones can be missed.

3.2.2 Morphology

Arabic has a complex morphology. Its derivational system is based on 10,000 independent roots, listed in a famous standard Arabic lexicon called لسان العرب (meaning: the Language of the Arabs) (Lisan Al-Arab, 2009), which is a very old thesuraus written by Ibn Manzour on the early period of Islam. Roots in Arabic are usually constructed from 3 consonants (tri-literals) and it is possible that 4 consonants (quad-literals) or 5 consonants (pent-literals) are used. But, the majority of roots (6,350 roots) are tri-literals (Moukdad, 2006). Out of the 10,000 roots, only about 1200 are still in use in the modern Arabic vocabulary (Hegazi and Elsharkawi, 1985). In English and many other languages, a root can be an

adjective, a noun or a verb. This is not the case in Arabic because roots in Arabic are verbs and verbs in Arabic are classified in three tenses: imperfective (present), perfective (past) and imperative (Attia, 2008). The base form of the verb is the perfective tense, 3rd person, singular. Thus all roots in Arabic are in this form.

Words and morphological variations are derived from roots using patterns. Patterns are used as standard frames for Arabic lexical words. Grammatically, the main pattern, which corresponds to the tri-literal root, is the pattern فعَل (transliterated as *f-à- l*). The pattern preserves "f", "à", and "l" in the same order. For example, in the tri-literal root حَسَب (meaning: to compute/count), "f" in the main pattern corresponds to the first letter "ح" in the root, "à" corresponds to the middle letter "س" and "ب" corresponds to the last letter "ل".

More regular patterns, adhering to well-known morphological rules, can be derived from the main pattern فعل (*f-à-l*). This is done by adding, according to rules, some letters initially, medially, finally or combinations of them. Examples of some patterns are فَعَل، فِعَال and أَفَاعِيل, transliterated as *f-à-l, f-i-à-l* and *a-f-à-i-l*, respectively. But, as the main pattern فعل (*f-à-l*) corresponds to a single root, the different patterns are also words derived from that root. Thus, the entire process is similar to a model in which original letters of a root are constants and variable letters are added to the root initially, medially, finally or in combinations, according to the patterns. In that context, it is possible to generate more than 20 derivatives from only one root. Table 3.6 on the next page illustrates several lexical words derived from the root حسب, which corresponds to the main pattern فعل (*f-à-l*), according to some different patterns, in which some letters are added to main pattern. Sometimes the new *patterned* word may have a totally different meaning when it is compared to its root meaning, e.g., the tri-literal root قتل means he killed while the patterned word قاتل, which is formed by adding the letter ALIF medially, means he fought.

Arabic Word	Pattern	Pattern Transliteration	Meaning
حسب	فعل	*f-à- l*	Compute (a tri-literal root)
يحسب	يفعل	*y- f-à- l*	He computes
حسبنا	فعلنا	*f-à- l-n-a*	We compute
حسبن	فعلن	*f-à- l-n*	They compute (plural feminine)
يحسبون	يفعلون	*y- f-à- l-o-n*	They compute (plural masculine)
حسبا	فعلا	*f-à- l-a*	They compute (dual masculine)
حاسوب	فاعول	*f-a-à-o- l*	Computer (Machine name)
حسّب	فعّل	*f-à- à- l*	He computes (for intensifying verbs)

TABLE 3.6: Different derivatives from the root حسب

Different kinds of affixes can be added to the derived patterned words to construct a more complex structure. Definite articles- like ال (its counterpart is the definite "*the*"), conjunctions, particles and other prefixes - can be affixed to the beginning of a word, whereas suffixes can be added to the end. For example, the word لنجْمَعنّهم (meaning: we will surely gather them) can be decomposed as follows:

(antefix: ل, prefix: ن , root: جمع , suffix: ن and postfix: هم). For the purpose of understanding stemming, all Arabic affixes are listed in Table 3.7, quoted in Kadri and Nie (2007).

Antefixes	Prefixes	Suffixes	Postfixes
وبال، وال، بال، فال، كال، ولل، ال، وب، ول، لل، فس، فب، فل، وس، ك، ف، ب، ل	ا، ن، ي، ت	تا، وا، ين، ون، ان، ات، تان، تين، يون، تما، تم، و، ي، ا، ن، ت، نا، تن	ي، ه، ك، كم، هم، نا، ها، تي، هن، كن، هما، كما
Prepositions meaning respectively: and with the, and the, with the, then the, as the, and to (for) the, the, and with, and to (for), then will, then with, then to (for), and will, as, then, and, with, to (for)	Letters meaning the conjugation person of verbs in the present tense	Terminations of conjugation for verbs and dual/plural/female/male marks for nouns	Pronouns meaning respectively: my, his, your, your , their, our, her, my, their, your, their, your

TABLE 3.7: Arabic affixes in MSA (Arabic is read from right to left).

Antefixes, whether they are separated or not, are usually prepositions added to the beginning of words before prefixes. Prefixes are attached to exemplify the present tense and imperative forms of verbs and usually consist of one letter. Suffixes are added to denote gender and number, for example in dual feminine and plural masculine. Postfixes are used to indicate pronouns and to represent the absent person. Usually this morphology is used to create verbal and nominal phrases.

Furthermore, the derivation form is influenced by Arabic syntactic rules (i.e. positions of words in sentences). For example, if the regular plural masculine lies in a nominative position in a sentence then it will have a different inflectional form from its position as an accusative, e.g., الموظفون (meaning: employees) as a subject and الموظفين as an object. For the purpose of understanding stemming, Table 3.8 summarises the entire process of how Arabic words are composed using the root كتب (meaning: he wrote). It is clear that this morphology results in a very large vocabulary. Ahmed (2000) stated that the estimated number of unique Arabic words (or surface forms) is 6×10^{10} words, cited in Darwish (2002a).

Steps in sequence	Arabic Word	English Counterpart	Notes
The original root	كتب	He wrote	A root can be: tri-literal, quad-literal and pent-literal
The pattern فِعَال (transliterated as *f-i-à-l,*)	كتاب	Book	Different patterns are used
Affixes are added	وكتابهم	And their book	Affixes include: antefixes, prefixes, suffixes and postfixes
Verbal phrases are constructed	وكتابهم بيدهم	And their book by their hands	Verbal and nominal phrases are used to construct sentences

TABLE 3.8: A typical sequence of steps for Arabic word construction.

Affixes in Arabic may include also some clitics. Clitics are morphemes that have the syntactic characteristics of a word but are morphologically bound to other words (Attia, 2008). Thus, clitics are attached to the beginning or end of words. Such clitics include some prepositions, definite articles, conjunctions, possessive pronouns, particles and pronouns. Examples of clitics are the letters ك and ف, which mean *as* and *then* respectively.

Morphology adds a level of ambiguity that makes the exact keyword matching mechanism inadequate for retrieval. Morphological ambiguity can appears in several cases. For example, the combination of clitics and words is a source of ambiguity in information retrieval because of the lack of boundaries between them. For example, clitics may accidentally produce a form that is homographic or homogenous (the same word with two or more different meanings) with another full word (Attia, 2007). For example, the word علم (meaning: science) can be joined with the clitic (ي) to construct the word علمي (meaning: my knowledge) which is homographic with the word علمي (meaning: scientific). In addition to these challenges, the flexible syntactic rules in Arabic lead to ambiguities in many cases. For example, a considerable deal of ambiguity is caused by the pro-drop nature of the Arabic language (Attia, 2008). The pro-drop nature means the subject can be omitted, leaving any syntactic parser with the challenge to decide whether or not there is an omitted pronoun in the subject position. For example, in the sentence ضرب الرجل, it is not clear if the word that follows the verb ضرب is a subject or an object. In the former case, meaning if the word is a subject, the sentence means 'the man hits' while the latter means 'the man was hit'.

Additionally, Arabic grammar contributes to the morphological ambiguity. For example, according to some Arabic grammar rules, sometimes vowels are removed from roots. The set of the vowel letters in Arabic consists of three letters: ALIF, YAA and WAW (أ، ي ، و). These letters have different rules that do not obey the derivational system of Arabic and make them very changeable. For instance, the last letter YAA is removed in a word like امشي (meaning: go), resulting in امش, if it appears in an imperative form. As another example, the last letter ALIF in the root نما (meaning: grew) will be modified to WAW in the present form of this root and thus it will be ينمو instead of ينما .

3.2.3 Diacritisation

Diacritisation, also known as *vocalisation*, refers to short vowels that appear above or below Arabic characters and words according to the Arabic parsing rules. Diacritisation is symbolised usually by superscript and subscript diacritical marks and can be embedded in words partially or fully. Diacritisation is used usually to disambiguate word meanings and to determine how words will be pronounced. In contemporary times, Arabic text is written without diacritical marks in news genre texts, formal letters and published articles and journals. This is typically a problem for non-Arabic speakers who represent more than 85% of Muslims (World's Muslim Population, 2013), while Arabic speakers can read text without any diacritisation. Diacritised texts are found in schools, linguistic lexicons and religious articles such as the Holy Quran and *Al-Hadith* (the teachings of Prophet Mohammad "may peace be upon him"). An example of an un-diacritised word is شعر. In this example the word شعر is very ambiguous unless it is

disambiguated using diacritical marks. For instance, the word شعر can be diacritised in different forms: شَعَر (meaning: he feels), شِعْر (meaning: poem) and شَعْر (meaning: hair).

Diacritisation in Arabic has three levels (Maamouri, et al., 2006; Habash and Rambow, 2007): vowel, shadda and nunation. Every Arabic letter can take any one of these diacritical levels. **Vowels** have four shapes written usually above or below letters. These vowels are *fatha*, *kasra*, *dhamma* and *sukun* (no vowel case). For example, the letter س, pronounced *SEEN*, can be diacritised in these four cases as follows: سَ in the word سَلام ; سِ in the word سِلْم ; سُ in the word سُكّر; and سْ in the word تَسْليم. These words mean *greetings*, *peace*, *sugar* and *delivery*, respectively. The sukun does not add anything to the written text and is only used for syllabic recognition in speech and is usually written as a small superscript zero shape. **Shadda**, known also as *gemination*, represents the implicit duplication of a letter, e.g., in the word العربيّة (meaning: Arabic) the shadda appears above the letter *YAA* (ي). **Nunation**, known also as *definiteness* or *Tanween*, consists of a short vowel plus an "n" sound and is attached usually to the end of nominals (nouns or non-verb words). Nunation can have three forms: *double fatha*, *double kasra* and *double dhama*. Unlike other types of diacritisation, nunation does not change meaning of words. A word in Arabic can take any form of this nunation according to its position in a given sentence as well as the parsing rules. For example, the word محدود (meaning: limited) can be *nunated* in the three forms of nunation as follows: محدودٌ, محدودٍ and محدوداً. Usually the form of double fatha adds a new *ALIF* letter to the end of words, for example in محدوداً in the last example.

Although diacritisation make words less ambiguous and more understandable, it is not used in current MSA. But, it was concluded that the absence of diacritics leads to a significant lexical and structural vagueness that cannot be disambiguated unless a contextual analysis is performed or at least readers should have adequate knowledge about the language's syntax and vocabulary in order to recognize the exact meaning. For instance, the word كتب (write) has more than 20 possible diacritised forms. Kamir, et al. (2002) illustrated ambiguity levels resulting from the lack of diacritisation with respect to linguistic levels, quoted in Abdelali (2006). An example of such ambiguity can result from indistinguishable meanings between words as a result of using the same un-diacritised lexical forms. e.g., the word ضرب may have different meanings without diacritics, '*he hits*, *a type* or *he was hit*'. Debili, et al. (2002), cited in Vergyri and Kirchhoff (2004), concluded that the non-diacritised dictionary word had 2.9 possible diacritised forms on average.

3.2.4 Broken Plural

Arabic has two types of plurals: broken and regular, which is known also as sound (Moukdad, 2006). The sound/regular plurals obey Arabic morphological rules, as in English (word and words). Since there is no neutral gender in Arabic, sound plurals have two sub-types: the sound masculine plural and the sound feminine plural. The sound masculine plural is constructed by adding the suffix ون to the base masculine noun in the nominative case, as in مهندسون (engineers), which is the plural of the word مهندس. In the accusative and genitive cases, the suffix ون is altered to ين as in مهندسين (engineers). The sound feminine

plural is constructed by dropping the suffix ه from feminine nouns in the nominative cases and adding ات at in its place. For example, the feminine plural مهندسات (meaning: feminine engineers) is the plural of the feminine noun مهندسه, which was itself constructed by adding the suffix ه to the base masculine noun. The suffix ات in feminine plural does not change in the accusative (like the case used for a noun when it is the direct object of a verb, e.g., me and him, in English) and genitive cases (like noun cases that are used to show possession in English).

In contrast, broken plurals in Arabic do not obey morphological rules. They are similar to cases like: *corpus* and *corpora*; and *mouse* and *mice* in English, but differing in that there is no rule-based morphological syntax to the broken plurals. Broken plurals constitute 10% of Arabic texts and 41% of plurals (Goweder, et al, 2005).Unlike English, the plural in Arabic indicates any number higher than two. The term broken means that the plural form does not resemble the original singular form. For example, the plural of the word نهر (meaning: river) is أنهار (rivers). In the simple cases of broken plurals, the new inflected plural has some letters in common when it is compared to the singular form, as in the previous example. But in many cases the plural is totally different from the original word, e.g., the plural of the word إمراة (meaning: woman) is نساء (women).

Diversity in broken plurals makes them highly unpredictable. In most cases knowing the singular form does not assist to deduce the plural, and vice-versa. This fact shows how much broken plurals lead to a mismatch problem in Arabic IR. According to Xu, et al., (2002) it seems that it would not be straight-forward to come up with a complete algorithm for generating broken plurals.

3.2.5 Synonyms

Arabic has a very rich vocabulary of synonyms. It also makes use of pseudo-synonyms - different words that describe different forms of a particular word or object. For example, in Arabic the lion (meaning: أسد) has more than 180 names (Fustat Adab, 2012) according to its age as well as its name's synonyms: (ليث, ضرغام, هيثم).

Synonyms produce a greater challenge to Arabic IR because it may lead to losing some relevant documents. The complexity of synonymy arises from the fact that Arabic has a special linguistic science called *ALBALAGHA* that encourages the use of synonyms and variable terms and phrases as an important feature for good speakers and writers. Goweder and De Roeck (2001), cited in Larkey, et al. (2007), stated that distributional analysis of Arabic newspapers showed that Arabic text has more words occurring only once and more distinct words than English text samples of comparable size.

3.3 Current Solutions to Monolingual Arabic IR

The above ambiguities have been solved to different levels in both Arabic monolingual and cross-lingual information retrieval. The next sections will shed some light on current solutions that have been adopted to address challenges and ambiguities in Arabic information retrieval.

3.3.1 Pre-processing and Stopwords removal

Preprocessing in Arabic includes removal of non-characters, normalisation of letters and removal of stopwords. Removal of non-characters (Abdelali, 2006) includes the removal of punctuation marks, diacritics and *Kasheeda*, known also as *Tatweel*, which is an Arabic stylistic elongation of some words for cosmetic writing. For example, the word عادل (a proper noun) can be written with *kasheeda* as عـــــادل. Normalisation in Arabic is used to render different forms of a letter with a single Unicode representation. This is important to moderate the orthographic variations. Such normalisation includes: replacing ALIF in HAMZA forms (*ALIF* combined with *HAMZA* that is written above or below the *ALIF* like in أ and إ) and *ALIF MADDA* (آ) with bare *ALIF* (ا); replacing final un-dotted *YAA* (ى) with dotted *YAA* (ي); replacing final *TAA MARBOOTA* (ة) with *HAA* (ه); replacing the sequence ىء with ئ; replacing the sequence يء with ئ and replacing ؤ with bare *ALIF* (ا). Most approaches pre-process documents and queries using some or all of these normalisations (Darwish and Oard, 2003b; Kadri and Nie, 2006). Abdelali, et al (2005) stated that some of these normalisations may conceal word characteristics and create ambiguity. In fact, such normalisations may hide regional variants, especially for transliterated words. For instance, it is not always correct to unify all glyphs of *ALIF* to a plain *ALIF* as it may lead to invalid words. Similar trends were also shown by Daoud and Hasan(2011) who showed that normalisation of Arabic letters, especially in the middle of words can result in errors. For instance, normalising ALIF MADDA (آ) with bare ALIF (ا) in the Arabic word قرآن (meaning: the Quran) results in the word قران (meaning: marriage contract).

To address the impact of Arabic challenges on both monolingual and cross-lingual retrieval and the problem of orthographic resolution errors, such as changing the letter *YAA* (ي) to the letter *ALIF MAKSURA* (ى) at the end of a word, the studies in Xu, et al. (2001) and Fraser, et al. (2002) used two different techniques to normalise spelling variations. The first technique is the *normalisation*, which replaces all occurrences of the diacritical *ALIF*, *HAMZA* (أ،إ) and *MADDA* (آ), with a bare *ALIF* (ا). The second technique is the *mapping*, which maps every word with a bare *ALIF* to a set of words that can potentially be written as that word by changing diacritical *ALIFs* to the plain *ALIF*. All the mapped words in the set are equally probable, each of which obtains *1/n* probability. The study of Xu and his team concluded that there is little difference between mapping techniques and normalisation techniques for orthographic resolution.

After normalisation, stopwords are removed. Examples of such stopwords in Arabic are إلى (meaning: to), من (meaning: from) and هو (meaning: him). A stopword list in Arabic includes words translated from English stopword lists, independent and dependent pronouns, normal and demonstrative prepositions plus a list of stopword phrases like السيد العزيز (dear Mr.). Stopwords may be used to determine the type of word that follows. Gey and Oard (2001), in their study of searching Arabic documents using English, French and Arabic queries, stated that the removal of stopwords is most commonly performed after stemming or morphological analysis in Arabic because the highly productive morphology of Arabic would otherwise result in impractically large stopword lists.

Some studies attempted to extract certain information from stopwords before their removal, as discussed in the previous chapter. Mansour, et al (2008) stated that it is possible to determine the POS of a word that follows a stopword. Some stopwords precede nouns only whereas others precede verbs. For instance, the conjunction لن (meaning: never) is followed usually by a verb while nouns follow prepositions only. A similar trend was also followed by Al-Shammari and Lin (2008a, 2008b), who divided the stopwords list into useful (those can classify the syntactic categorization of the subsequent words) and useless (those give no benefit to the immediately following words). This was done to determine the appropriate stemmer for stemming Arabic words, as will be discussed in the stemming section in this chapter.

Regardless of the used approach, most existing methods use dictionaries, compiled lists or software tools, in which the stopwords list can be easily manipulated and accessed, to determine which words are to be removed.

3.3.2 Tokenisation

Tokenisation is used intensively when building corpora or preparing for POS tagging e.g., building a multilingual parallel corpus for Arabic-Spanish-English by Samy, et al.,(2006). The simple method to tokenize Arabic texts is to use the white space delimiter. However, as Arabic has complex morphology, the process is not that simple. Accordingly, different morphological analysers have been developed for Arabic tokenisation and information retrieval. Buckwalter (2002) developed a stem-based morphological analyser for tokenising MSA words.

The Buckwalter algorithm is one of the most popular and respected analysers. It is based on three groups (prefixes, possible stems and suffixes) and three valid combinations in the form of truth tables: prefix/stem pairs, prefix/suffix pairs and stem/suffix pairs. Buckwalter, coded in a program called AraMorph, takes Arabic words with or without short vowels as an input and makes morphological analysis and POS tagging along with a list of possible translations.

The Buckwalter algorithm divided every input word into three sub-strings with all its possibilities. If the first sub-string is a legitimate prefix, the second sub-string is a legitimate stem, the third sub-string is a legitimate sub-string and if the combination of all of them is valid then the second sub-string will exemplify the stem of the input word. If more than one stem is obtained then all of them will be listed. For example, for the word تعمل (*tEml* in Buckwalter transliteration), a version of the Buckwalter analyser provided many solutions - two of them are presented in Figure 3.2 on the next page.

In this figure, one of the selected solutions – specifically SOLUTION #1- for the word تعمل (meaning: it/she/they work/works) is grammatically categorised in two parts, the first is the letter TAA ت with the diacritic fatha (resulting in تَ) (transliterated in Buckwalter as: ta and translated as it/they/she). The second part of the word is classified as the verb عمل (transliterated in Buckwalter as: Eomal and translated as work/function/act).

```
Processing token : تعمل
Transliteration :  tEml

SOLUTION #1
Lemma  :  Eamil
Vocalized as :     taEomal
Morphology :
          prefix : IVPref-hy-ta
          stem : IV
          suffix : Suff-0
Grammatical category :
          prefix : ta        IV3FS
          stem : Eomal       VERB_IMPERFECT
Glossed as :
          prefix : it/they/she
          stem : work/function/act

SOLUTION #2
Lemma  :  taEam~ul
Vocalized as :     taEam~ul
```

FIG. 3.2: Two solutions for the word تعمل (meaing: she works/you work) using the Buckwalter analyzer.

One deficiency of Buckwalter's analyzer is that some words may not be stemmed because they may not be included in the stem table. In addition, broken plurals are not managed by the Buckwalter stemmer (Xu, et al., 2001). Attia (2008) lists 11 cases where the Buckwalter analyzer failed to get their stems. Some of the listed shortcomings are: Buckwalter failed to stem clitic question morpheme because of lack of coverage for such cases, e.g., أعادل (meaning: *Is it correct that Adil*); and Buckwalter lacks in the coverage of imperative forms, and, according to Attia's evaluation, out of 9198 verbs only 22 verbs (0.002%) have imperative forms. This shortcoming limits Buckwalter from dealing with instruction manuals as usually such manuals include verbs in imperative forms; Buckwalter is limited in the coverage of the passive morphology; and Buckwalter is unable to handle multi-word expressions, like مجلس الأمن (meaning: Security Council), which are very prevalent in MSA. Multiword expressions produce ambiguous words because analyzing them as separate units leads to the loss of their meaning.

Diab, et al. (2004) developed different Arabic morphological software to solve difficulties in tokenisation, POS tagging and Base Phrase Chunking of MSA. In their solutions, Diab, and her team utilised a supervised learning approach that uses training data from the Arabic Tree Bank and is based on using SVM (support vector machines). The Diab's tokeniser strips the prefix letter و (meaning: *and*), some definite articles like ال as well as some suffixes, such as possessive pronouns and normal pronouns. In their results, Diab, and her colleagues stated that their tokeniser scores 99% in accuracy while their POS demonstrates 95.49% accuracy. In addition, the developers concluded that their approach is language independent. One advantageous point for the analyser of Diab and her colleagues is that it has the ability to strip some prepositions that cannot be removed by other analyzers and stemmers (Larkey, et al., 2007). In contrast, Larkey and her team (2007), in their experiment on Arabic stemming, concluded that

Diab's analyser has many mistaken tokenisations in morphological analysis when it was supplied with sample AFP articles (News articles in Arabic). For example, the tokeniser sometimes did not separate the definite article ال (meaning: the). Such mistakes prevented words from getting the correct POS.

Attia (2007, 2008) evaluated previous projects in Arabic tokenisation and he concluded that none of these projects have the ability to handle and disambiguate many words and multi-word expressions. The evaluation also concluded that ambiguity in some morphological analysers, like Buckwalter, is caused by some features like the inclusion of classical entries, rule-created over-generated stems with no actual place in the language, and ignorance of some word-clitic combination rules. Attia enumerated sources of ambiguities that may arise due to syntax ambiguities in Arabic: the pro-drop nature of the language, word order flexibility, lack of diacritics, and the multi-functionality of Arabic nouns. Therefore, Attia described an ambiguity-controlled rule-based tokeniser model, based on finite-state technology. The developer aimed to control morphological and syntactic ambiguities in Arabic. The analyser handles the problem of ambiguity at different levels by different phases: pre-processing (tokenisation, morphological analysis or POS tagging) and parsing (like phrase structure rules, lexical specifications). Attia stated that using such rules makes ambiguities much more manageable. Some of Attia's models include: tokenisation combined with morphological analysis; tokenisation guesser for clitics; and tokenising multiword expressions. Results, which were tested on short sentences extracted randomly from a news corpus, showed that the Attia analyser outperforms many analysers such as Buckwalter's. In addition, his parser achieved 92% coverage of grammar (complete parses) after applying the above-mentioned techniques. This is relatively high coverage compared to other languages. For example, according to the developer, the German parser, which was an LFG-based (Lexical Functional Grammar) parser as the Attia's one, produced full parses in approximately 84.5% of full sentences. The concluding point about Attia's models is that these models have the ability to resolve ambiguity.

3.3.3 Stemming

Since Arabic is an inflectional language, a large number of studies have been devoted to the analysis of the best approach to index Arabic words. As in stemming in other languages, Arabic stemming techniques can be also classified into the two major techniques: root-based (heavy or morphological analysis based stemming) techniques and light stemming-based (affix removal stemming) techniques (see section 2.2.2.3 in the previous chapter). For example, root-based algorithms produce the root عمل for the word وأعمالهم (meaning: and their works) because prefixes, suffixes and infixes are removed while light stemmers generate أعمال because only prefixes and suffixes are removed.

The Khoja stemmer (Khoja and Garside, 1999) is one of the most famous algorithms in Arabic stemming. It is a root-based algorithm that produces the roots of words to stem by removing the longest prefixes and the longest suffixes. The algorithm was widely used in Arabic IR literature. One advantages of the algorithm is that it can detect letters that were deleted during the derivation process, such as in امش and نما (illustrated at the end of section 3.2.2), and returns these removed letters back to the root. Nevertheless, Khoja's stemmer has some drawbacks. For example, Khoja may result in an over-stemming

problem (see section 2.2.2.3). For instance, Khoja produces the root طفل (meaning: child) for the word طفيليات (meaning: parasites), and produces the root لَعِب (meaning: play) for the word لعوب (meaning: irresponsible), which are totally different in meaning. Such failures occur because many words with different meanings in Arabic may have the same root. Furthermore, the algorithm is not always able to deal with the clitics problem and proper nouns as well, although the algorithm is based on a mechanism that returns foreign words unchanged. This mechanism is based on the fact that some words do not have roots. Hence, if Khoja's stemmer comes across any of these words, it returns the original word. In addition, sometimes the algorithm removes some affixes that are parts of words. In his study for the Holy Quran, Hammo (2009) stated that most of the failing cases of Khoja when it was used to stem words of the Holy book, were due to stemming proper names such as the names of Prophets, angels, ancient cities, places and people, numerals, as well as words with the diacritical mark, shadda.

Darwish (2002a) developed *SEBAWAI,* an Arabic analyzer that is based on automatically derived rules and statistics. *SEBAWAI* has two main modules. The first module constructs a list of "word-root" pairs i.e (سحب , وسحابهم). Then it extracts a list of prefixes, suffixes and stem templates and follows this process by estimating the probability that a prefix, suffix or stem template would occur. For example, given the pair (سحب ,وسحابهم) in the example above, the system produces و (meaning: and) as the prefix, هم (meaning: theirs) as the suffix and "CCAC" as the stem template (C's represent the letters in the root).The second module of *SEBAWAI* takes a word and produces the possible combinations among prefix, suffix and template. These combinations are obtained by eliminating prefixes and suffixes from words and then comparing all the produced stems to templates. As a result, a list of ranked roots is produced. These roots will be matched automatically against the list of the 10,000 roots extracted from an electronic copy of Lisan Al-Arab (Lisan Al-Arab, 2009) to confirm their existence. *SEBAWAI* has some limitations stated by its developer. First, it cannot stem transliterated words such as entity names because it binds the choice of roots to a fixed set. Second, *SEBAWAI* is not able to deal with words that have one letter length. Such words are very limited in Arabic, e.g., عِ (meaning: grasp). Third, *SEBAWAI* cannot deal with some individual words that constitute complete sentences, like لَنَهْدِيَنَّهُم (meaning: we will surely guide them).

To mitigate against such types of losing stem semantics, light stemming for Arabic was also used (see section 2.2.2.3). In Arabic IR, studies indicated that light stemming outperforms root-based approaches (Aljlayl and Frieder, 2002). The rationale is that the latter conflates more terms, which degrades the performance. Nevertheless, light stemming techniques may result in an under-stemming problem (see stemming section in previous chapter). For instance, broken plurals do not get conflated with their singular forms because they preserve some affixes and internal differences. In spite of its shortcomings, it is shown that there is no sophisticated approach that is more effective than light stemming for Arabic (Larkey, et al., 2007).

Al-stem is a light stemmer, presented by Darwish (2002b), which lightly chops off the following prefixes (وال، فال، بال، بت، يت، لت، مت، وت، ست، نت، بم، لم، وم، كم، فم، ال، لل، في، وا، وا، فا، لا،با) plus the following suffixes (ات، وا، ون، وه، ان، تي، ته، تم، كم، هم، هن، ها، ية، تك، نا، ين، يه، ة، هـ، ي، ا). Darwish and Oard (2003b) used *Al-stem* in their experiment to develop a technique for combining evidence for Arabic-English cross-language information retrieval at TREC 2002. The experiment is presented later in this chapter.

Based on the assumption that light stemming preserves the meaning of words, unlike root-based techniques, Aljlayl and Frieder (2002) proposed an algorithm to stem Arabic words lightly. The algorithm strips the most prevalent suffixes (i.e. possessive pronouns), prefixes (i.e. definite articles) and any antefixes or postfixes that can be attached to the beginning of the prefixes or the end of suffixes. Aljlayl and Frieder, however, did not list their removable sets of prefixes and suffixes explicitly. In some cases the algorithm uses a normalisation technique for words as well as remove all the diacritical marks except the *shadda*, because it is a sign for a duplication process of a consonant and thus *shadda* exemplifies a letter that could be lost if shadda is removed. One advantage of the algorithm is that it can deal with some arabicised words according to a predefined list. However, entries in such a list would probably be limited in its coverage. Aljlayl and Frieder concluded that their light stemming algorithm outperforms root-based algorithms, in particular the Khoja stemmer.

Larkey, et al., (2002) proposed several light stemmers (*light1*, *light2*, *light3* and *light8*) based on heuristics and some strippable prefixes and suffixes. The affixes to be removed are listed in Table 3.9. In the implementation, the algorithms of these different versions of light stemming perform the following steps: peel away the letter و (meaning: and) from the beginning of words for *light2*, *light3*, and *light8* only if there are 3 or more remaining letters after removing the و (such condition avoids removing words that start with the letter و); truncate definite articles if this leaves 2 letters or more; and remove suffixes, listed in table below from right to left, from the end of words if this leaves 2 letters or more.

Light stemmer type	Removing from front	Removing from end
Light1	ال، وال، بال، كال، فال	None
Light2	ال، وال، بال، كال، فال، و	None
Light3	ال، وال، بال، كال، فال، و	هـ ، ة
Light8	ال، وال، بال، كال، فال، و	ها، ان، ات، ون، ين، يه، ية، هـ، ة، ي

TABLE 3.9: Strippable strings removed in light10 stemmer.

In monolingual and cross lingual experiments, developers of *light8* concluded that it outperforms the Khoja stemmer, especially after removing stopwords with or without query expansion. Later, Larkey, et al., (2007) expanded their previous study by adding another light stemmer called *light10*. *Light10* removes the following prefixes from the beginning of words (ال، وال، بال، كال، فال، لل، و) and the following suffixes from the end (ها، ان، ات، ون، ين، يه، ية، هـ، ة، ي). The only difference is the addition of (لل) for strippable strings from the beginning. The hypothesis is to truncate frequent prefixes and suffixes that are infrequently found at the beginnings or endings of words. In their monolingual and cross lingual experiments, developers showed that *light10* outperforms their previous light stemmers as well as the Khoja stemmer and results showed that *light10* is better even when query expansion techniques are not implemented. In particular, the developers of *light10* stated that it is far better than Khoja. Results also indicated that root-based stemmers, like Khoja, are probably not the best techniques for stemming in

cross-lingual retrieval, if the translation of the root of each query word was used checked in a dictionary. This is because the root will obtain too many translations and most of them are incorrect.

In the same study, Larkey, et al. (2007) compared *light10* to slightly modified versions of the Buckwalter and Diab analysers, which were introduced earlier in tokenisation in section 3.3.2. Larkey, Ballesteros and Connell concluded that *light10* with stopword removal outperformed the Buckwalter and Diab analysers, including their modified versions, if queries are not expanded. With query expansion, the Buckwalter stemmer is equivalent to the *light10* stemmer. *Light10* has become a fashionable solution to stemming and has been added to the Lemur toolkit[14].

Xu, et al., (2001) implemented a stemming approach based on the Buckwalter analyser. In particular, the developers used two techniques: *sure-stem* and *all-stems*. With the *sure-stem* technique, a word is stemmed if it has exactly one stem. If a given word does not have a stem, then it will be left as is. With the *all-stems* technique, a word is resolved to all its possible stems probabilistically, with the assumption that all stems are equally probable because there is no training data but later a probabilistic IR model will handle such ambiguity. Results showed an improvement on the recall of over 10% when it is compared with full-word stem. Results also showed that *sure-stem* is somewhat better than *all-stems*, but the improvement is not statistically significant. Abdelali (2006) concluded that their approach may fail to eliminate ambiguous words. Since the same probability is assigned to both valid stem and possible stems, noise may be introduced.

The same authors extended their study to include spelling normalisation (Xu, et al., 2002). In that study, spelling normalisation (variants in spelling) was implemented to detect the confused cases of some letters (i.e. YAA and ALIF). In the experiments, Xu and his colleagues concluded that the use of spelling normalisation for orthographic variation with 3-grams and stemming improves Arabic retrieval performance significantly by 40%. Surprisingly, in this experiment, Xu and his colleagues stated that stemming and spelling normalisation have a small impact on cross language information retrieval, unlike the results by the developers of the *light10* stemmer, who used the same TREC 2001 data. With respect to stemming, Larkey, et al., (2002) explained that Xu, et al. (2001), used a parallel corpus, extracted from a UN corpus, so their bilingual lexicon contains all the variants of Arabic words. However, Larkey, Ballesteros and Connell used a bilingual lexicon derived from an online dictionary, so it contains fewer variants. This means that query terms were not matched against the dictionary entries unless they were stemmed.

Stemmers based on corpus statistics, known as statistical stemmers (see stemming section in CLIR review chapter), were also explored. Mustafa and Al-Radaideh (2004) stated that the use of a di-gram method for Arabic information retrieval offers better performance than tri-grams with respect to precision and recall ratio. In their algorithm, Mustafa and Al-Radaideh implemented word-based stemming and concluded that the N-gram method is not an effective solution to corpus-based Arabic word conflation.

Experiments from TREC 2001 and TREC 2002 (Gey and Oard, 2001; Oard and Gey, 2002) in both Arabic monolingual and cross-lingual retrieval are among the most important experiments. Different techniques used for indexing terms. Examples include n-grams and root-based stemming. Results showed that

[14] http://www.lemurproject.org/

further investigation is needed from Arabic IR systems. The experiments and their results are discussed, in terms of cross language information retrieval, later in section 3.5.

Chen and Gey (2002) implemented a new approach for Arabic stemming using statistical stemmers that use parallel corpora. Chen and Gey used an English stemmer to stem English words in an English-Arabic parallel corpus. Then, Arabic words are clustered together into a stem category depending on their mapping to English stems in the corpus after being aligned and processed with GIZA++, which was described in the previous chapter. Results showed that the increase in performance was substantial when it was compared with Al-stem.

Xu, et al. (2002) combined Arabic monolingual N-gram retrieval with stemmed words. The study showed that the use of tri-grams combined with stemming improved retrieval, though this improvement is not statistically significant. The study also experimented with bi-grams and di-grams, instead of tri-grams. Results indicate that both of them do not outperform tri-grams because bi-grams are very short with little context while di-grams are similar to word or stem-based retrieval.

Inspired by the drawbacks of both light and heavy stemming techniques, Kadri and Nie (2006) proposed a new stemming technique known as linguistic-stem. The developers employed Arabic morphological rules to produce all the stem's candidates. Then, the most appropriate stem will be selected depending on corpus statistics. The used corpus was obtained from the Arabic TREC collection. Every word in the corpus was decomposed to produce all the possible stems in the collection for that word. Thus, a corpus of stems with their occurrence frequencies was built.

In the same study of linguistic stemming, Kadri presented his light stemming approach, for comparison reasons, which is somewhat similar to the ones mentioned above and shares several prefixes and suffixes with them. Kadri created a statistics table based on the occurrence frequencies of both prefixes and suffixes on 523,359 different tokens in the TREC collection. By using this statistics table, Kadri sets the most frequent prefixes and suffixes in order to be removed. The judgment to remove a prefix or a suffix is taken according to some rules and statistics on the corpus also. Kadri and Nie concluded that their algorithm is better than the common light stemmers, in particular Kadri's light stemmer. However, linguistic stemming depends on the corpus statistics, meaning that it may still lead to some errors.

Mansour et al. (2008) presented an auto-indexing approach to build indices for Arabic documents. In their indexing process, the algorithm firstly tagged every word into verbs and nouns using morphological rules. The process was managed by a set of predefined rhythms (patterns). Secondly, the algorithm removes stopword and stop-list phrases. Thirdly, the algorithm identifies nouns and verbs depending on the preceding word, as it illustrated in the stemming section in this chapter. Fourthly, the algorithm extracts stems from the rhymed/patterned words. In particular, some morphological rules were used to extract stems from both nouns and verbs. For instance, verbs were checked firstly against some exceptional grammatical rules for Arabic verbs. If such scenario fails, then verbs are checked against the "*ten-verb-additions*" rule (grammarians of Arabic stated that the derivative system of any verb has 10 known different formats) after being heavily investigated to remove non-essential letters and thus the stem of any verb is obtained. Finally, Mansour and his colleagues assign weights to the stemmed words relative to their documents, depending on some statistical factors like the frequency of occurrence of a

word in its containing document. Thus, all the possible stems of a word will be sorted according to their weights. Developers concluded that their method is very useful and obtain an average recall of 46% and an average precision of 64% when it is tested with 24 arbitrarily selected general-purpose texts with various lengths.

Al-Shammari and Lin (2008a, 2008b) proposed a novel algorithm for stemming Arabic words, known as Educated Text Stemmer (ETS), by the use of syntactic knowledge and morphology. The hypothesis of the study is that in Arabic there are some useful stopwords, which can be used to identify verbs and nouns. Accordingly, the Khoja stemmer was applied to stem verbs while light stemming is applied to nouns. Using two samples of data, in particular 47 medical documents with 9435 words and 10 sports articles (7071 words), Al-Shammari and Lin evaluated their educated text stemmer. They concluded that their stemmer was able to generate 96% correct stems. In addition, they stated that the ETS stemmer produces better results when more documents are contained in the stemming process.

Daoud and Hasan(2011) claimed that most existing techniques for Arabic stemming are not adequate due to its highly inflectional feature. For example, it is not always feasible to determine if the letter(s) is a suffix or a prefix, e.g., كتابان (meaning: two books) or it is a part of the stem like in أثمان (meaning: prices) because words boundaries are not always clear. For instance the last letter in the latter word is a part of the stem, although the light 10 stemmer would remove it. Furthermore, most algorithms, heavy or light, depend solely on removal of longest match affixes as the first step. For instance, for the strings واثبات (meaning: proof or confirmation 'attached to and'), light 10 stemmer produces the stem اثب , which is an awkward word. Accordingly, Daoud and Hasan(2011) begins the stemming process by the stem, rather than the removal of the affixes. This is done by segmenting the Arabic word (or string) according to a lookup dictionary that contains only valid stems. In that context, the longest match is returned whenever number of words is minimized. Daoud and Hasan(2011) claimed that their algorithm is the ideal stemmer when it was compared to both Khoja and light 10 stemmers. Furthermore, they found that Khoja is stronger than light 10. However, the results should be interpreted in their context. This is especially true because the sample text used in the experiments contains only 8697 distinct words. Words, even if they are synonyms, may belong to different root words. Current stemming techniques, however, are not able to conflate such synonyms to the same stem class. Inspired by this drawback, Mohamed, et al. (2011) proposed a new technique for Arabic document retrieval using Wikipedia. The key idea was based on collecting concepts with their synonyms in a dictionary from the downloadable dumped database of the Arabic Wikipedia, in which redirect pages usually represent other different names (abbreviations, synonyms, etc) for concepts (articles). Accordingly, documents, after being tokenized, are processed in term of n-gram and if a particular n-gram matches any of the synonyms of a certain concept, then the term would be substituted by its right concepts, which are demonstrated by their synonyms, for example, in the concept dictionary. Using Arabic TREC-2001 with Arabic queries, results showed that the effect of using such an approach was not better than stemming techniques, but it is hoped that the continuous growth of Wikipedia may result in changing this effectiveness tendency towards concept-based IR.

3.4 Arabic-Specific Techniques

Unlike most western languages, Arabic has extra challenges to IR due to its rich morphology. Challenges like broken plural and regional variations were investigated by the IR community and it has been proven that solutions to such challenges improve retrieval effectiveness. The next section discusses these techniques in Arabic IR.

3.4.1 Broken Plural Resolution

Several techniques were proposed to deal with the broken plurals problem. The problem is mainly related to monolingual retrieval. For CLIR, it is not a key problem because plurals and singulars can be translated separately (Xu, et al., 2001). In particular, Xu, et al. (2001) proposed a statistical-based thesaurus, extracted from the UN English-Arabic parallel corpus, to deal with the large number of broken plurals and synonyms in Arabic. The key idea was based on the fact that English translation of both singular and plural forms of the same Arabic word would likely be stemmed to a single English stem. Hence, the problem collapsed to a synonymy problem, in which synonymous Arabic words, even if they are in singular or plural forms, would be translated to the same word in English. The experiment is discussed in terms of CLIR in section 3.5.1.2; however, results showed that the automatically derived thesaurus was very useful due to the fact that most broken plurals were identified successfully by the thesaurus.

Since singular form and broken plural form have some common letters in many broken plurals, the use of character n-grams to detect the broken plural is one of the solutions that were proposed (Xu, et al., 2002). In this approach, the developers implemented n-grams created from stems as well as n-grams from words. Results concluded that stemming by the use of n-grams with the stemmed word is better than n-grams with the word-base. The reason behind is that some of the word-based n-grams are prefixes or suffixes. Details of these results were discussed earlier in the stemming section in this chapter. Xu's team claimed that there may not be a straight-forward algorithm to handle the broken plural in Arabic.

Goweder, et al. (2004) and Goweder, et al. (2005) proposed three approaches for identifying broken plurals. The first approach is the *simple broken plural matching*, which matches the light-stemmed words with 39 pre-defined patterns, e.g., أفعال (transliterated as: *a-f-à-a-l*), of broken plurals, extracted from standard Arabic grammar references to determine whether the word is a broken plural or not. For example, the broken plural الأقلام (meaning: pens) is lightly stemmed as أقلام, without the definite. The pattern of this word is أفعال (transliterated as: *a-f-à-a-l*, which is one of the broken plural patterns). This approach scores low precision - approaching 13% - on a test set of about 187,000 words. The reason behind this low precision is that broken plural patterns are too general to obtain good performance.

In order to restrict broken plural patterns, the same developers proposed the second approach (Goweder, et al., 2004; Goweder, et al., 2005), which they named *restricted broken plural matching*. In this approach, a set of rules was extracted from 18.5 million words so as to control broken plural applicability. First the developers stemmed words lightly. The output of stemming - about 127,000 words

- was fed to the simple broken plural matching, which was explained above, to obtain all stems that match broken plural patterns. Next, words that were classified in a specific broken plural were examined in order to produce a restricted pattern. For example, both the words أغبياء (meaning: broken masculine plural for stupid) and احتواء (meaning: inclusion) have the same broken plural pattern: أفعلاء (transliterated as a-*f-à-l-A-a*). So, in order to reject the word احتواء (inclusion) and to accept the word أغبياء (stupids) as a broken plural, the developers examined both patterns and concluded a rule stating that if the word, which is tied to a broken plural pattern, has the *ALIF* (أ) in position 1 and 5, the *HAMZA* (ء) in position 6 and the letters in positions 2-3-and 4 matches the tri-literal root pattern فعل (*f-à- l*) - illustrated in morphology in section 3.2.2- , then the word is a broken plural. Several such rules were extracted. In the experiments, developers extracted restricted patterns manually, on the first run, and automatically by using a supervised machine learning technique. Next, the developed broken plural identification approaches were incorporated in some light stemming approach. Results, which were conducted using a test collection of 187,000 words, showed that the overall performance of automatic restricted patterns increased, reaching about 75% in precision.

The third approach for identifying broken plural was built on the top of the previous approaches of matching (Goweder, et al., 2004; Goweder, et al., 2005). The approach used a dictionary which lists broken plural stems. This dictionary was constructed automatically by extracting all instances of broken plural stems that match broken plural patterns. Next, sets of rules, as in the previous approach, were extracted. Results showed that a significant improvement in precision, reaching 92%, over the other two approaches was obtained.

3.4.2 Regional Variations

There is a continuum of spoken dialects varying geographically, but also by social class, which are native languages. These dialects differ phonologically, lexically, morphologically and syntactically from one another (Habash and Rambow, 2006) and in all Arabic countries speakers usually used a hybrid approach between their dialects and the MSA. This results in a very regional vocabulary. Abdelali (2006) discussed regional variations in MSA. He presented an approach to improve Information Retrieval by restricting the semantics of the words used within a variation. The semantic information was inferred from linguistic resources formed by using language modeling techniques. Based on language variation from language identifiers, Abdelali built up a shorter index per variant (region) by classifying the input documents. A similar query classification will limit the search into its corresponding index variant, for a much quicker and precise search. Semantic closeness is represented by semantic vectors, based on vector space models. A “T matrix” is used to form semantic vectors by recording the number of co-occurrences of terms in one window of specific size. Then, the SVD method was used to reduce the size of the matrix T. Arabic regional variations that were covered by Abdelali include Levantine Arabic, Gulf Arabic, Egyptian Arabic and North-African Arabic. Abdelali showed experimentally that his method is promising and resulted in a significant increase in recall and precision over peer systems, which use query expansion techniques. Results also showed that the use of small but semantically related numbers of

words/documents in the expansion is better that adding too many words. The latter approach may hurt retrieval effectiveness.

3.5 Arabic Cross-Language Information Retrieval

Arabic-English CLIR, as an example of Arabic CLIR, allows users to find relevant documents written in English, whereas their queries are issued in Arabic. Most of the current approaches to Arabic CLIR systems are based on those discussed in the previous chapter. Sections 3.5.1 to 3.5.3 illustrate some of these approaches to Arabic CLIR.

3.5.1 Translation Approaches

As in CLIR, the dominant approach in Arabic CLIR is query-based translation using bilingual MRD, MT and parallel corpora.

3.5.1.1 Dictionaries and Machine Translation

Aljlayl and Frieder (2001) conducted a study of Arabic-English CLIR using both MRD and MT. They adopted three methods for the MRD: *Every-Match* (EM), which uses all the translation candidates; *First-Match* (FM), in which only the first translation is considered; and *Two-Phases* (TP), which is a bi-directional translation with no probabilistic knowledge, as discussed in bi-directional translation disambiguation section. Using TREC data and these three methods, Aljlayl and Fireder concluded that the TP method outperforms the EM and FM methods for Arabic-English CLIR. Similar conclusions were also shown in the previous chapter in section 2.2.4.2. In the machine translation experiment, Aljlayl and Frieder used commercial MT software and they concluded that translation results were better than the Every-Match method (EM) but both the FM and TP methods outperformed the machine translation approach. In another study, Aljlayl, et al. (2002) investigated also the effect of context on the quality of translation by using various query lengths. Results showed that if fewer source terms are needed to form a context, the retrieval accuracy and efficiency is better. In addition, the study concluded that MT systems do not achieve high quality translation because they cannot distinguish between ambiguous cases that are lexical and ambiguous syntactic cases.

In TREC 2001 and TREC 2002 experiments, ten groups explored the effect of using different sources of translation knowledge for Arabic CLIR (Gey and Oard, 2001; Oard and Gey, 2002). The purpose of the experiments was to search Arabic documents using original English queries and their translations in both Arabic and French. Several mechanisms were implemented for monolingual and cross-lingual runs. Table 3.10 on the next page summarises the different indexing terms, the query languages, and the sources of translation, for cross-language runs. As the table shows, four indexing techniques were used: word, stem, root and n-gram. The word was a surface form obtained by tokenisation at white space and punctuation.

The stem was obtained by the light removal of both prefixes and suffixes, such as light-8 stemmer and other stemming software. The n-gram was used with values of *n* ranging from 3–6. For the translation resources of queries, the experiments also implemented four approaches: Machine translation approach, which used two MT systems; translation lexicons, which used three commercial machine readable bilingual dictionaries; parallel corpus, which was obtained from the UN documents; and a pronunciation-based transliteration. All the ten participating teams implemented a "bag-of-words" technique based on indexing statistics about the occurrences of terms in each document. A wide variety of specific techniques were used for retrieval, including language models, vector space models and inference networks.

Team	Arabic Terms Indexed				Query Language	Translation Resources Used			
	Word	Stem	Root	n-gram		MT	Lexicon	Corpus	Translit
BBN		X			A, E	X	X	X	
Hummingbird		X			A				
IIT	X	X	X		A, E	X	X		
JHU-APL	X			X	A, E, F	X			
NMSU	X	X			A, E		X		
Queens	X			X	A, E	X			
UC Berkeley		X			A, E	X	X		
U Maryland	X	X	X	X	A, E	X			X
U Mass	X	X			A, E	X	X		
U Sheffield	X				A, E, F	X			

TABLE 3.10: Techniques used by participating teams for Arabic IR in TREC 2001 (Translit: Transliteration, A: Arabic, E: English, F: French).

The experiments highlight three themes (Abdelali, 2006): (1) A greater focus on exploring innovative CLIR techniques than was evident in TREC-2001; (2) continued investigation of Arabic-specific issues, such as stemming and stopword removal; and (3) increasing reliance on multiple sources of evidence to overcome the limitation of any single source (Oard and Gey, 2002).

Larkey, et al. (2002, 2007) conducted a study to evaluate the effect of different stemming approaches in Arabic cross-lingual and monolingual IR. The study used English queries against the TREC-2001 Arabic corpus. Using the dictionary approach, all the Arabic translations for an English word were grouped together and they were treated as instances of the original term (structured with SQM). For the Arabic-English dictionary, the studies used some lexicons gathered from several online English-Arabic and Arabic-English resources on the Web and had been implemented also by Larkey and Connell (2001). The study concluded that root-based stemming is probably not efficient for Arabic-English CLIR if the source for translation is based on dictionary look-up because too many translations are obtained, most of them being semantically different. In addition, it was concluded that the *light10* stemmer scores the best result for Arabic-English CLIR, whether queries are expanded or not.

Levow, et al. (2005) tested different stemming/matching techniques in an English-Arabic CLIR experiment using TREC-2002. Techniques involved the use of white-space delimited tokens, linguistic stems, root stems and lightly stemmed words. The translation was performed using a bilingual term list extracted from some Internet sources. Structured query model was also applied whenever more than one translation candidate was produced. Results showed that, among the different used techniques, light stemming was the most effective algorithm as Arabic is significantly affected with adequate, rather than heavy, morphological handling. Results were consistent with those obtained in Larkey's studies. For example, the probabilistic structured query (Darwish and Oard, 2003), in which several translation resources were used to estimate translations probabilities, was originally developed using the Arabic TREC-2002 collection with English queries, as was discussed in the probabilistic structured query section. The results showed that the technique is very effective and it can be utilized for many other languages.

Xu, et al. (2001) combined two sources for estimating Arabic term translation probabilities: a manual lexicon (obtained from three different sources) and a parallel corpus (obtained from the UN documents). For the manual lexicon, uniform translation probabilities are assumed for the English translations, that is, if an Arabic word has n English translations, each translation gets probability 1/n. Results concluded that the word ambiguity problem in Arabic is satisfactorily handled by complementing a manual lexicon with a parallel corpus.

3.5.1.2 Parallel Corpora

Parallel corpora were also used in Arabic CLIR experiments. For example, Xu, et al. (2001) in TREC 2001 implemented a statistical thesaurus to address the problems of broken plurals and synonyms in Arabic, as illustrated earlier in section 3.4.1. The thesaurus was taken from the UN parallel corpus and from a manual lexicon, containing word pairs obtained from different sources. A simple alignment algorithm was used to align sentences. GIZA++ with "IBM Model 1" was used for the estimation of the translation probabilities and for the lexicon extraction. The experiment showed that the automatically extracted thesaurus improved the overall performance, almost certainly because of the resolved broken plurals. In TREC 2002 CLIR, Fraser, et al. (2002) used the same approach of a statistical thesaurus but GIZA++ was used with "IBM Model 4" for the lexicon extraction. However, results were at the same level of improvement of the experiments that were conducted by the same authors in TREC 2001.

3.5.2 Transliteration and OOV

The task of transliteration becomes more challenging when the language pair uses different orthographies, such as in the case of English and Arabic. Beside those types of variants discussed in section 3.2.1, in Arabic the different orthographic variants of transliterated foreign/cross-linguistic/Romanised names contribute to making transliteration more challenging. For instance, some of the work in Abdelali, et al. (2005) discussed some issues related to the Arabic corpus newswire AFP, like inconsistency in translation/transliteration. For instance, that work identifies different variants in Arabic

for the city Los Angeles (لوس أنجلوس). Unfortunately, Arabic culture contributes to the variability of foreign/cross-linguistic names in Arabic and makes it chaotic. In Arabic literary tradition there is a famous proverb that says "*It is a foreign name you can write it any way*" (Arabic Proverbs, 2013). Thus foreign names in Arabic can be written in any way. As an example, Table 3.11 documents three different hits returned by the Google search engine (Google, 2013), using various Arabic spellings of the name England. The search was conducted in February 2013. Such a problem is very prevalent in Arabic texts.

Variant	Number of hits
انجلترا	9,420,000
انكلترا	3,320,000
انجلتره	114,000

TABLE 3.11: Number of hits for the Arabic counterparts of the word 'England'.

Some studies attempted to improve retrieval of Arabic names by constructing a large database and using statistical methods (i.e. frequency and classic Levenshtein algorithm, which is a fuzzy matching technique used to measure similarity between two strings using the edit distance. The edit distance can be identified by the total number of insertions, deletions and substitutions that are needed to make two strings similar to each others). Salhi and Yahya (2011) developed a set of tools for Arabic people names processing and retrieval. For examples, tools include gender detection and translation from Arabic to English. Most methods were primarily based on simple frequencies and mapping from a database collected from students' lists of the Palestinian General High School Certificate for the time period 2005 to 2010 and some other resources as well. Beside the translation counterpart in English and the name's gender, the database also includes many compound names like عبد الرحمن (Abd ElRahman). The database can be considered as a valuable resource for Arabic proper nouns. But, yet many other Arabic people names are not included and thus transliteration techniques must be used.

Contrary to the above case of Arabic words that originated in English, English/Romanised words may have different variants if they originated in Arabic. Kashani, et al. (2007) identified 87 different and official English spellings for the name of the ex-Libyan leader Muammar Gaddafi "معمر القذافي". As was previously mention, a similar observation was also noted by Whitaker who found 32 different English variants for the name of the ex-Libyan leader, cited in AbdulJaleel and Larkey (2003). Reasons behind these different orthographic variations in English or in Romanised Arabic are listed in AbdulJaleel and Larkey (2002, 2003) and Zawaydeh and Saadi (2006). First, Arabic and English/Romanised letters are not in a one to one correspondence. In fact, there may be different phonetics between the two languages, e.g., Arabic does not have a "*p*" letter while "*S*" may have several mappings in Arabic. Second, English spelling is irregular. Third, short vowels in English do not have correspondences in Arabic. However, short vowels (diacritical marks) in Arabic are exemplified by vowel letters in English.

AbdulJaleel and Larkey (2002) proposed a simple statistical method to automatically learn a transliteration model from a sample of name pairs in Arabic and English. Their purpose was to reduce

the effect of OOV by transliterating proper nouns and place names from English to Arabic since more than 50% of OOV words are named entities, according to analysis done by the developers. The method contains two models: monogram transliteration model and bigram transliteration model. The monogram model is a group of probability distributions over context-independent English and Arabic characters. So each English letter can be transliterated into an Arabic letter with a conditional probability, e.g., *Pr* (س|s) = *x* while *Pr* (ز|s) = *y*. Under certain conditions English or Arabic letters may take a null (zero probability) due to different lengths of words. Sometimes two English letters may be transliterated into one Arabic letter. For instance, "*ph*" together are transliterated into one Arabic letter "ف" (*FA*). Conversely, some English letters may be transliterated to two different letters in Arabic, as in the letter "*h*" that can be transliterated to "هـ" or "ح" (*HAA* and *HHA* respectively). To handle such cases of letters the method makes use of a bigram transliteration model. For the character-level alignment phase, AbdulJaleel and Larkey used GIZA++ (Och and Ney, 2003). GIZA++ was trained on a parallel list containing person names and place names in Arabic and English. Results indicated that the model's accuracy depends on the size of the corpus used. In addition, bigrams are more effective than monograms. This approach is able to produce multiple alternative Arabic spellings and thus variations in Arabic are covered.

AbdulJaleel and Larkey (2003) extended their previous proposed system. Similar to their previous study, a parallel list containing English proper nouns and their translations in Arabic was used for the training. The list was obtained from different online and off-line sources. The training model was built in different steps. The model starts by normalising every Arabic and English word in lists. Then, words were segmented into unigrams and GIZA++ was used to align the *Arabic-English* word pairs with Arabic as the source language. If there are cases in which a sequence of English characters is mapped to a single Arabic character, then the 50 most frequent of these sequences were added to an English inventory. After that the English words were assembled again according to the new English inventory that was created in the previous step. Next, GIZA++ was used again to align *English- Arabic* word pairs with English as the source language. Finally, the alignments from the GIZA++ output were counted and converted to conditional probabilities. Thus, a transliterated Arabic word is produced by segmenting an English word according to the n-grams inventory. For each segment, all the possible transliterations are generated. Each word transliteration gets a score in order to get its rank. The developers named their method 'the selected n-gram model'. AbdulJaleel and Larkey found that transliteration for the OOV terms improves CLIR performance and produces a significant increase in precision, especially when query expansion techniques are implemented, while adding transliteration for all named entities did not significantly improve performance. This means that it is better to transliterate only words or names that do not already have translations in the dictionary. Furthermore, the selected n-gram model method is more accurate than the unigram model.

3.5.3 Query Expansion

Query expansion techniques were also used in Arabic CLIR. Xu, et al. (2001) performed sequential expansion by expanding English queries in TREC 2001, and then the expanded English queries were used to retrieve the top documents for the purpose of expanding Arabic queries. They showed that such expansion can propagate English expansion errors to Arabic expansions. Their result showed that query expansion was effective for both Arabic monolingual and cross-lingual IR. In fact, in the experiment, Arabic query expansion alone showed better performance than using both Arabic and English expansion together. The possible reason behind this, as stated by the researchers, is that the weights for English expansion terms are larger than they should be. The same team implemented independent query expansion techniques (parallel expansion) for both English and Arabic in TREC 2002 (Fraser, et al., 2002). For an English query expansion, they used a corpus of 1.2 million articles. The AFP corpus and additional articles were used for the Arabic expansion. 50 terms were selected from the top 10 retrieved documents. The experiment showed that their results in TREC 2002 did not improve retrieval performance, although different methods, including the use of parallel expansion instead of sequential expansion, were explored.

In another experiment, Xu, et al. (2002) used local feedback with an Arabic thesaurus derived from the (UN) parallel corpus, to select 50 terms from the top-10 retrieved documents based on their total TF×IDF scores to expand queries for Arabic monolingual information retrieval. Xu's team showed that merging both feedback and thesauri together outperformed (by 15%) the use of feedback alone. The result is statistically significant. The team concluded that feedback and thesaurus use are two different techniques for query expansion.

Several query expansion techniques for Arabic CLIR were investigated in TREC 2001 and TREC 2002 experiments (Gey and Oard, 2001; Oard and Gey, 2002), which were already discussed in section 3.5.1.1. These expansion techniques were implemented by several participants' teams. The techniques include: pre-translation feedback, post-translation feedback, sequential expansion, parallel expansion, document expansion and the Rocchio approach to blind feedback.

Darwish and Oard (2003b) performed an experiment for expanding both query and document for both Arabic monolingual and Arabic-English cross-lingual IR. Their experiment was performed using pre-translation query expansion. In each document, Darwish and Oard first extracted the 20 most expressive terms. They did this by dividing the frequency with which each term appeared in the document by the number of documents in which that term was found. Then they composed a query with one instance of each of those 20 terms and used that query as a basis for ranking the documents in the AFP collection using an IR system. Later, Darwish and Oard combined the 10 top-ranked documents into a single *mega-document* and then the 20 most descriptive terms in that mega-document were selected, using the same measure of term importance as described above. The resulting set of 20 terms was then added to the representation of document that was being expanded. Darwish and Oard used document expansion in the monolingual run while the pre-translation query expansion was done using blind relevance feedback for Arabic-English CLIR. Results showed that further work on document expansion is needed.

Larkey, et al. (2007) expanded Arabic queries combined with the use of different stemmers using the technique of local context analysis to add 50 terms from the top 10 documents. In the CLIR run, the experiment showed that the best results for average precision was achieved when both Arabic and English queries were expanded.

Abdelali, et al. (2007) stated that wrong selection of expansion degrades and biases the retrieval process. Therefore, they presented a query expansion mechanism that has the ability to automatically select a corpus related semantically to the query. In particular, instead of using matching semantics "word-by-word" between query words and words from a corpus, the developers used a query vector to find the closest matching set of word/document vectors in a corpus. Later, the cosine similarity measure was used for the matching process. The approach utilises Latent Semantic Analysis (LSA) for an effective expansion. The study, which was done using the Arabic TREC-10 data, showed that this approach is promising. With word expansion, the mechanism retrieved more relevant documents and achieved a 5% increase in number of relevant documents retrieved over the baseline run without query expansion. With document expansion, the approach achieved an additional 60% increase in number of relevant documents and over 98% higher precision when it is compared to word expansion.

3.6 Summary

This chapter presents three major parts in both Arabic monolingual and cross-lingual information retrieval: the major components of Arabic morphology that shape its retrieval and how they affect ambiguities in IR; the essential ingredients in Arabic IR; and the current approaches and solutions that have been implemented to tackle Arabic IR, either monolingual or cross-lingual. It can be concluded that Arabic IR still requires deep exploration since the optimal solutions and effective Arabic IR systems for both monolingual and cross-lingual are still distant. The rationale behind this is reflected in different points: pre-processing techniques are not unified yet; each of root-based stemming and light stemming has its pros and cons and it is not clear whether linguistic-based stemming is more appropriate; algorithms used in retrieving Arabic documents in current search engines still need strong morphological rules; broken plural and regional variation need much research; and adapted solutions for Arabic cross-lingual CLIR with its translation resources, transliteration and query expansions are still far from optimal. Additional challenge to Arabic CLIR is the issue of mixed-language querying. It was shown that non-Arabic speakers are not always able to express terminology in their native language. Furthermore, it can be seen that the most Arabic CLIR approaches are also focused on monolingual weighting and retrieval, even the queries are translated. Motivated by these facts, this thesis attempts to experimentally develop algorithms for language-aware IR (mixed-language IR) systems. It also attempts to address the issue of bilingual mixed querying in CLIR. It was shown that a large number of non-English documents, including Arabic ones, usually contain many words/phrases in a secondary language, mostly English. Such words often co-occur in bilingual forms. Furthermore, words can be written in different languages in the same documents. Such features usually cause the CLIR system to provide biased list towards mixed documents.

Accordingly, this thesis attempts to incorporate mixed-language feature in queries and documents in the computations of TF, DF and document length components. It attempts to moderate most drawbacks, e.g., overweighting, that were discussed before in the introduction chapter with regards to mixed-language feature in queries and documents. Details of proposed approaches to mixed-language IR systems are described in the next chapter.

CHAPTER

4

Mixed-Language Information Retrieval

At its highest level, this thesis attempts to contribute to the development of current algorithms and approaches that can search multilingual and mixed data collections as well as to address the problem of multilingual querying, which stems from users' ability to express certain terminology/concepts in a particular language that is different from their own. In particular, the major problem the thesis attempts to solve is how to produce the most relevant documents, regardless of their language(s) or the dominant language in the query words, at the top of a retrieved ranked list whenever a user search is posted in a multilingual/mixed form (multilingual querying).

With this challenge in mind, this chapter presents the algorithms and the solutions that were designed and introduced to better suit the unique characteristics of this mixed-language problem in both queries and documents. The suggested approaches are centered on two key components in the IR task, weighting and the indexing. In particular, in a centralized index, a new model, specifically a variant of structured query models, for estimating TF, DF and document length was developed, whereas in a traditional distributed architecture a new architecture for indexing documents in multilingual and mixed collection and a model for re-weighting documents in such type of indices were proposed. However, the solution models of weighting and indexing differ in the type of the employed indexing architecture.

Before delving into each solution's details, the chapter first provides complete examples for the major shortcomings, which were identified in the introductory chapter, of each architecture and why it is no longer applicable to IR task with mixed-language queries and documents. Following this, the proposed solutions for weighting and/or indexing of mixed documents and queries are presented. Although the proposed algorithms are being focused on Arabic as a primary language, with English as a secondary

language in queries and documents on common computer science vocabulary, the developed approaches are designed to enhance information retrieval for any language that is not in widespread use and whose vocabulary does not span modern terminology e.g., computer science and technology.

Section 4.1 introduces the mixed-language matching problem. Section 4.2 illustrates the major limitations that were previously identified, with explanatory examples, in the centralized architecture whenever it is utilized for indexing and weighting mixed documents and queries. It also introduces the weighting model that is proposed as a solution for mixed-language documents and queries in a centralized index. The section shows the mathematics of how the three major parameters in weighting (term frequency, document frequency and document length) are estimated using several suggested methods and why it is essential to utilize a decaying factor in such estimations in mixed queries. Furthermore, the section also describes another proposed re-weighting method, specifically re-weighting the IDF so as to handle traditional overweighting while on the same time to minimize overweighting caused by mixture of texts. Additionally, section 4.2 shows how the two proposed re-weighting can be combined. Section 4.3 is focused on the distributed architecture (mixed-language in separate indices). It firstly proposes a new indexing approach for multilingual IR. Secondly, the section introduces a probabilistic approach for term weighting in mixed documents. Finally in section 4.4 the chapter is concluded.

4.1 The Problem of Mixed-Languages Matching

It was shown that CLIR allows users to search documents that are written in a language different from the query, but matching queries in multiple languages with documents in monolingual and/or mixed languages is a different task. This is mainly because direct matching usually biases the result list towards mixed documents, whereas many monolingual and highly relevant documents can be easily missed, as described in the first chapter of this thesis. This suggests that the linguistic disparity between monolingual documents and mixed query, whereas it is not between mixed documents and mixed query, makes monolingual and mixed documents incomparable in their weightings whenever a mixed query is posted. This is especially true because most IR systems depend on frequencies and document statistics of terms, regardless of their languages. With such an assumption, matching is often performed exactly with no consideration to languages, resulting in incomparable scores between mixed and monolingual documents, as the scores of the latter documents would be computed from only a portion of the mixed query.

This incomparability between monolingual and mixed documents in mixed-languages matching, especially in technical jargon, stem mainly from two difficulties: how both monolingual and mixed documents/queries are indexed and how they are weighted and ranked.

On one hand, indexing of mixed-languages documents (besides the monolingual ones) is a major issue since an appropriate architecture would have to be either determined or designed, whenever, a mixed-language IR system is to be developed. It was shown that the underlying assumption is that a typical CLIR (and also MLIR) task reduces the matching process, between documents and queries, to a

translation followed by a monolingual retrieval. Accordingly, most existing indexing strategies are designed for indexing several monolingual documents, rather than mixed documents with two or more languages. As described previously, mixed documents in existing CLIR and MLIR approaches are deliberately/cautiously ignored as they are often handled as if they are written in a monolingual language.

On the other hand, the same grounding belief of monolingualism makes mixed-languages weighting stands also as another difficulty through the matching process, as the majority of the current weighing methods are particularly optimized for monolingual queries and retrieval, rather than mixed. Therefore, most approaches seek to disambiguate translations, when several alternative (but monolingual) translations are identified using a translation resource. In that context, weighting takes place implicitly during the translation disambiguation process, resulting in integrated techniques that compromise both the weighting and the translation with an essential assumption, that is, the query set is monolingual. In that perspective, queries are usually translated from query language to document language.

It is true that there are some approaches that perform the translation bi-directionally, meaning from query language to document language and vice versa, as described in section 1.1.2. These approaches, however, are different to the work presented here in several points. First, queries in these different approaches were essentially monolinguals with a grounding base that the test collection is monolingual too and the major aim is to disambiguate translation, rather than handling mixed-language feature in queries and documents. Second, the different approaches were mostly developed on the top of news-genres test collections. The conclusion, however, about ranking function of news domain documents using these approaches does not mean that these algorithms are readily to be applied in other genres such as specialized computer science domain. There is always a possibility of undesirable behavior and/or poor performance once moving from one domain to another domain. Third, the special characteristics of mixed-language in both queries and documents, e.g., the feature of the co-occurrence of terms in different languages in mixed documents, in these different approaches are almost ignored or not handled adequately.

With these trends in mind, the next sections describe proposed approaches in terms of centralized and distributed indexing approaches.

4.2 Mixed-Languages in a Unified Index

The centralized indexing architecture, in which all documents are stored in a single index, is the major approach in CLIR (and also in some MLIR), as illustrated in section 2.3. In CLIR, the centralized architecture of indexing is implicitly assumed. Given the premise that a CLIR task is a monolingual retrieval preceded by a translation, the reason for this principal implicit assumption (meaning the use of one index) is that documents in document collection are monolingual and, thus, they will be placed in a single index. In traditional MLIR, the centralized architecture is intensively used, as it attempts to facilitate the task of indexing by avoiding the merging problem, as illustrated earlier.

At first glance, the centralized architecture appears adequate for indexing multilingual documents, because of making use of a single index. This assumption makes sense and is appealing in its simplicity for mixed documents, but yet the centralized index has been shown to have some drawbacks, in general, and for mixed-language queries and documents, in particular, as described in section 1.1.2.1. Such drawbacks include monolingual overweighting, overweighting due to mixture of texts, biased TF due to the occurrence of the same term together in different languages (bilingual co-occurring terms), biased DF due to independent computations of terms that are cross-lingually similar and the dominance of the mixed documents on top of the retrieved result list. These drawbacks make the utilization of the centralized architecture at least problematic and it is not the optimal solution for indexing mixed documents (and monolingual ones, as well), unless weighting is modified. The next explanatory example shows these drawbacks in terms of weighting in the centralized indexing architecture.

4.2.1 Illustrative Example

Consider the illustrative example that follows. The example is selected to reflect problems that were listed in the previous section. The example will be used through this chapter and most proposed approaches will be applied to this sample corpus to make comparison easier.

A multilingual query Q = ' مفهوم الـ Inheritance' (meaning: concept of inheritance) is posted to a multilingual document collection containing 14 documents with 7 documents in English, 3 documents in monolingual Arabic and 4 documents mixed (in both Arabic and English). The collection consists of the following documents:

D_1: "يدعم الفكرة الأساسية لإعادة استخدام البرامج **inheritance** مفهوم الـ"

D_2: " تسمح بإنشاء تصنيفات هرمية **Inheritance** فكرة **الوارثة**"

D_3: "The concept of **inheritance** allows the creation of hierarchical classifications"

D_4: "Java does not support the **inheritance** of multiple superclasses into a subclass. This is different from **inheritance** in C++. Unlike **inheritance** in C++.."

D_5 : "**Inheritance** is one of the cornerstones of object-oriented programming. Using **inheritance** you can create a general class that…."

D_6: "تؤثر **الوراثة** بشدة على تعريف المتغيرات. لذلك فإنها"

D_7: " على خصائص إضافية من كائن آخر. لذلك فـ**الوراثة** تعني أن يتحصل كائن object **الوارثة** **Inheritance**"

D_8: " تستخدم **Inheritance** فكرة تنظيم البرامج المعقدة..... ذلك بسبب أن الـ **Inheritance** تدعم **الوراثة**"

D_9: "**الوراثة** البشرية الموجودة للابن كما هو الحال في مفهوم**الوراثة** تقوم بنقل كل الخصائص"

D_{10}: "**Inheritance** supports reusability… **Inheritance** of general attributes... Using **inheritance** mechanism makes it possible to add general attributes…. However, **inheritance**"

Besides these 10 documents, there exist 3 irrelevant documents in English and 1 irrelevant document in Arabic.

In this collection, D_2 and D_3 are identical, as D_2 is the exact Arabic translation of D_3. However, since D_2 is in Arabic, the translated English term 'inheritance' co-occurs with its Arabic equivalent 'الوراثة'. As Arabic

script is written cursively from right to left these two co-occurred terms, meaning 'inheritance' and 'الوراثة' , appearing at distance places. Technically, if D_2 is written in Arabic, its presence would likely be:

"مفهوم الوارثة Inheritance يسمح بإنشاء تصنيفات هرمية"

In which the two terms 'inheritance' and 'الوراثة' are neighbours. The same phenomenon appears also in both D_7 and D_8.

Table 4.1 illustrates the document similarity computations when the multilingual query Q is translated, concatenated to form a single query and then submitted to our multilingual collection. For simplicity, computations are provided for the keywords: 'Inheritance' and its translation 'الوراثة' only. Similarity is computed in terms of the standard $TF_{i,D}$* IDF_i formula with *ntn.ntn* weighting scheme (in which $TF_{i,D}$ denotes the number of occurrences of term *i* in document *D* and IDF_i is the IDF factor of term *i*). The *ntn.ntn* weighting scheme is based on SMART notation for TF-IDF variants (Manning, et al., 2008). The first letter 'n', which is the abbreviation of the word natural, in each triplet in this weighting scheme refers to the use of natural term frequency component. The second letter 't' in each triplet refers to the use of the IDF, whereas the third letter 'n', which is stands for none, refers to that no normalization is used for the document length (Manning, et al., 2008).

Docs	inheritance	الوراثة	Documents' scores
	TF * IDF	TF * IDF	
D_1	1 * 0.24304	0* 0.44716	0.05907
D_2	1* 0.24304	1* 0.44716	0.25902
D_3	1* 0.24304	0* 0.44716	0.05907
D_4	3* 0.24304	0* 0.44716	0.17721
D_5	2* 0.24304	0* 0.44716	0.11814
D_6	0* 0.24304	1* 0.44716	0.19995
D_7	1* 0.24304	2* 0.44716	0.45897
D_8	2* 0.24304	1* 0.44716	0.31809
D_9	0* 0.24304	2* 0.44716	0.39990
D_{10}	4* 0.24304	0* 0.44716	0.23627
Q	0.24304	0.44716	-

TABLE 4.1: Computations of ranking in the sample collection for the query 'مفهوم الـ Inheritance'.

The DF and IDF for the term 'inheritance' are 8 and $log(14/8) = 0.24304$, respectively, while the DF and IDF for the term 'الوراثة' are 5 and $log(14/5) = 0.44716$, respectively. According to these computations, the ranking of documents would be D_7, D_9, D_8, D_2, D_{10}, D_6, D_4, D_5, D_1 and D_3. However, D_1 and D_3 have the same scores.

First of all, it is notable that the difference in scores between D_2 and D_3 is disappointing, although both documents are identical. This primarily results from two causes. Firstly, because the Arabic term 'الوراثة' tends to co-occur with its equivalent English term in D_2, the document earns double weights, one for

each term. This is the biased TF problem in which weights are distorted. Secondly, the overweighting due to mixture, the traditional overweighting as well as the independent computation of weights for the terms 'الوراثة' and 'inheritance' cause the former Arabic term to earn additional weight represented in its term distribution statistics (IDF). However, a one might say that this due to the high DF, which is 8, of the English term – compared to the Arabic one, which is 5. This is true but, it is mainly related to what is found in retrieval of realistic environment. In such environments the total number of English documents, especially in scientific domains, is yet much greater the numbers of any non-English documents, including those in Arabic and in mixed languages. In fact, in the example the total numbers of both Arabic monolingual documents and mixed documents are much higher compared to retrieval in realistic environment.

The findings in Table 4.1 also give evidence that the top ranked list is dominated by those documents that exactly contain the same terms in the multilingual query, resulting in the presence of the most mixed documents in the first four ranks. Beside the previous mentioned causes in the above paragraph, again this is due to the fact that the query attributes higher importance to the Arabic translation 'الوراثة' than the English term 'inheritance'. Another observation from the table is that although D_{10} is highly relevant document, at least in terms of TF, in the collection, it is ranked at the middle of the retrieved list. This is an undesirable trait in which monolingual English documents can be easily missed due to the partial matching with the mixed query. In particular, a significant portion of the multilingual query would not match the monolingual documents, e.g., D_{10}because their weights are computed from only small portions of the mixed queries, unlike mixed documents, whose weights are computed from the entire multilingual queries. However, this is not the case for monolingual Arabic documents, which compensate this by their biased IDF factor. This is why D_9, which is a monolingual Arabic document, gets higher rank, particularly rank 2. It is also observed that D_4 is ranked at the lowest half of the result list, despite of its relevance.

Obviously, the attributes of such weighting in this example is not desirable. Accordingly, managing such shortcomings is crucial to improve the accuracy of term weighting in mixed documents.

4.2.2 Cross-Lingual Structured Query Model

Most ranked retrieval models in IR depend primarily on similarity ranking methods that are based solely on term frequency, document frequency and document length components. A weight is then assigned to each term that appears in the query, using these listed statistics, so as to compute similarity coefficient scores between the query and documents. These scores are then employed to obtain the final measure of the document relevance.

With respect to mixed queries and documents that use a centralized index, the weight component is an essential facet that should be controlled carefully. In particular, it is important to eliminate problems such as biased TF, dominance of mixed documents on top and overweighting whether it is traditional or caused by mixture of texts and biased DF.

Recall the common feature of mixed documents. A given technical term in a certain language, e.g., Arabic, may frequently appear in the same mixed document but in multiple languages and at two or

more distant positions, sometimes even not in the same vicinity. Such features can be utilized to propose reasonable weights for terms in mixed queries. In particular, the intuition implies it is convenient to assume that both the query term and its equivalent translation(s) are synonyms but across languages (in two different languages). That is, the presence of one translation(s) of a given source query term in the document impacts - in terms of weights - the presence of that query term. For a query that requests the source term, a document with just one or more of its translation(s) along with any source term should be retrieved. The major subsequent result to this general assumption is that weights of such synonymous terms across languages would likely be computed together as if these terms are a single similar term, rather than decomposing their computations individually. Thus far the mixed document can be viewed as if it is in a single monolingual language and resulting in making monolingual documents comparable and more competitive to those mixed ones. Using such a paradigm makes the weight cross-lingual, instead of monolingual.

This is the first step on the notion of the proposed approach, which is called the *cross-lingual structured query model*. The model is targeting both the TF and the DF, as in most variants of the structured query model, in mixed documents but it also adds the document length component L when the weights are assigned to terms in mixed queries.

However, before delving into how these three factors are estimated in mixed documents, it is important to bear in mind how a centralized architecture formulates queries. A source query is just attached to its translated version. However, queries in this work are multilingual in their origins – as they are posted by users. Therefore, the source mixed queries are partially and bi-directionally translated. Partial and bi-directional translation here means that the English portion in the multilingual query is translated to Arabic and vice-versa. The end product of this bi-directional translation is two monolingual queries, one is in Arabic and the second is in English, which will be merged as in traditional centralized index. However, since usually the English portion in multilingual queries is assumed to be a technical term and expected to be highly significant, as discussed earlier, its translation is obtained using an in-house-built and special dictionary on common computer science vocabulary. With respect to the Arabic snippet(s) in mixed queries, it is often taken from the general-purpose vocabulary. If it is not a stopword, its translation is obtained using other translation resources after being matched, firstly, with the entries of the special dictionary, whose vocabulary is inverted in the second direction (from Arabic to English). Meanwhile, in the proposed modification on weights (cross-lingual structured model) the information of whether a certain translation is obtained from the special dictionary or not, is the measuring criterion for applying or ignoring the proposed weight modification. This is because translations from an special dictionary is precise and specific, whereas translations of terms that are taken from general purpose vocabulary are often general and may skew result list. Accordingly, each translation word is associated with its certainty as follows: if the translation of a source query term is obtained from the computer dictionary then this source term and its translation(s) are re-weighted according to the proposed weight that follows; otherwise original weights of both the source query and its translation(s) are kept. The issue of how mixed queries are bi-directionally translated and how the special dictionary is used are detailed in the experiment chapter. Nevertheless, the language (whether it is in Arabic or in English) of terms in the

mixed merged query is referred in this chapter as the source language query term whereas the language(s) that appears in documents, regardless of the one that presents, is referred to as the target language(s).

4.2.2.1 Initial Estimation of Term Frequency

In the first stage of the proposed re-weighting method, the intuitive step is to specifically target how often synonymous terms across languages occur in a particular mixed document in order to suppress the biased TF problem. This is based on the frequencies with which of each synonym across languages. Consider a mixed document D in which the frequency of the source query language term q is $TF_{q,D}$, whereas in other distant positions the same document contains some translation A_i for the same source query language term with $TF_{i,D}$ as the number of occurrences. In such a mixed document it is reasonable to handle these terms as synonymous terms across both the Arabic and the English languages. According to this assumption, if the source query language term q or its translation A_i appears in the mixed document D, these terms are treated as if the query term q occurs in the document D and hence, both the translation A_i and the term q are considered as synonyms but in different languages. Apparently, the TF in such a case can be considered as a variant of the TF in the structured query. However, it is a cross-lingual structured in this case. Formally, this cross-lingual TF of the source query term can be expressed as follows:

$$\grave{TF}_{q,D} = TF_{q,D} + \sum_{\{i|i \in T_q\}} TF_{i,D} \tag{4.1}$$

Where, D is a mixed document in more than one language. However, it can be also a monolingual document, $\grave{TF}_{q,D}$ is the new computed frequency of occurrences of the source term q (this is the joint TF of synonyms across languages), the $TF_{q,D}$ is the frequency of occurrence of the source term q in the document D, T_q is the set of the translations of the term q in the target (document) monolingual language and $TF_{i,D}$ is the number of occurrences of a given element in the set T_q that appears in the document D. Thus, $TF_{T,D}$ is the number of occurrences of the terms in the set T_q that occurs in document D. The symbols are derived from Levow et al. (2005).

4.2.2.2 Decaying the Term Frequency of Co-occurred Terms

Based on features of mixed documents that have been shown before, a scientific Arabic term in such documents is often accompanied by its corresponding translation(s) in English. The phenomenon of such co-occurrences is usually found in forms of neighbouring, e.g., deadlock الإقفال, or very closed pairs, in which terms appear with some words between them. For example in the sentence key المفتاح بأنواعه the word *key* is separated from its translation المفتاح with one word. Accordingly, the words can be defined to be very closed to each other. Although the co-occurrence feature has been employed and used intensively

by numerous studies for several purposes in CLIR, e.g., OOV resolution, it may undesirably increase the weight of mixed documents and cause them to earn additional values to their scores. Thus, the more co-occurred pairs in a document, the more it can skew result list. Recall the explanatory example above. A typical problem is approached due to co-occurrence of terms across languages in that example when it was found that there is a significant difference between the weight of D_2 (0.25902) and the weight of D_3 (0.05907), despite the fact that both document D_2 and document D_3 are identical.

It is observed in formula 4.1 that the frequency of each term, which appears in a co-occurred pair in a mixed document (and also in mixed merged query) is counted no matter whether these terms are presented in the mixed documents independently or they occurred together (co-occurring terms in different languages). Therefore, the second step of the proposed weighting aims to circumvent this problem. The premise made here is that since a source term tends to co-occur with its equivalent translation or vice versa, especially when documents address technical topics, it is unlikely to compute weights for each of the co-occurring terms (a weight for the term in Arabic and another weight for its translation in English) because this would result in double computation in a mixed document and, thus, the result list becomes biased towards mixed documents. In such a case, it is reasonable to apply a reasonable decaying factor for the TF of terms across languages based on how frequently these terms co-occur together in mixed documents. This is what is called the *decaying factor*.

To estimate this decaying factor, the frequency of a particular co-occurred pair is used, that is when a source term co-occurs together with its translation(s) or vice versa, the contribution of this bilingual co-occurrence is rebalanced by decreasing the frequency of the source query term (which is computed as the sum of the TFs of terms across languages as in equation 4.1) by 1. This attribute will result in an overall decaying value that is equivalent to the count of the frequencies of the corresponding co-occurred pair. Obviously, in proportion the higher number of occurrences of a certain pair, the less its value in the new computed TF. The technique can be viewed as a smoothing-like mechanism whose role is to damp or down scale the TF contribution of terms that participate in co-occurring terms.

It is important also to note bilingual terms across languages are considered as 'co-occurred terms' in this thesis if they appear together in a window of size 5. Inside this window, however, any occurrence of two bilingual terms is handled without considering their appearance order (which term appears first and which appears second). Such an assumption seems reasonable for two situations. Firstly, terms don't need to be in the same order in texts, e.g., deadlock الإقفال and الأقفال deadlock are similar. Secondly, individual terms in phrases did not need be exactly neighbouring to each other, e.g., the cross-lingual phrase 'mutual exclusion الاحتكار المتناوب'. Such cross-lingual phrases are prevalent in non-English documents. Thus, the decaying factor for cross-lingual term frequency estimation was applied only when co-occurred bilingual terms are found in any order, within a window of 5 words. At this point, let's assume that the source language query term q is placed in a set Q and its translations are placed in another set T_q, where $T_q = \{a1, a2,\ldots, an\}$. The Cartesian product between these two sets will generate the possible pair combinations between each source query term in the source query language with one of its translations in terms of pairs, e.g., (q, a_1), $(q, a_2)\ldots$, (q, a_n). Hence, the decaying factor of TF of the source query (synonyms across languages) can be extended to formula 4.1 to result in:

$$C_q = Q \times Tq \tag{4.2}$$

$$\grave{TF}_{q,D} = TF_{q,D} + \sum_{\{i|i \in T_q\}} TF_{i,D} - \sum_{\{n|n \in C_q\}} TF_{n,D} \tag{4.3}$$

where C_q is the resulting set from the Cartesian product between the set Q, which only contains one element, and the set T_q, the $TF_{n,D}$ is the frequency of occurrences of each element (pair) in the set C_q in document D. Summing up the number of occurrences of all pairs in the set C represents the decaying factor. Other terms in the formulae are previously defined.

Docs	Synonyms (inheritance and الوراثة)	
	Joint TF	Decaying value
D_1	1	0
D_2	2	1
D_3	1	0
D_4	3	0
D_5	2	0
D_6	1	0
D_7	3	1
D_8	3	1
D_9	2	0
D_{10}	4	0

TABLE 4.2: The joint TF for the reference collection example.

If the earlier example is recalled, Table 4.2 illustrates both the joint TF (the sum up of TF) and the used decaying factor in each document in the retrieved list. It is important to note that the specificity of the co-occurrence of Arabic-English pairs depends on the documents' authors. Some authors keep on writing the two pairs together whenever a certain Arabic technical term appears, while others write them frequently after every portion talks about the topic of the term. Others write them only on titles and in a few distant positions. In scientific Arabic documents all these cases exist.

4.2.2.3 Estimating Document Length

Document length is an essential component in similarity computations. This is because the longer the documents the more terms paired with distinguished terms are assumed to be found and consequently leading such documents to have higher TF as well as increasing the likelihood of containing terms that match the user's query.

Indeed, modification of TF in equation 4.2 has a potential consequent impact, even if it is low, on the document length. Since the TF of the co-occurrence of Arabic-English bilingual pairs is modified, it is commonplace to reflect this update on the number of terms in mixed documents. Obviously, the

document length should be decreased also by 1 whenever each co-occurred term pair (which are computed across languages) appears in the retrieved mixed document. Formally:

$$\grave{L_D} = L_D - \sum_{\{n|n \in C_q\}} TF_{n,D} \quad (4.4)$$

Where $\grave{L_D}$ is the new length of document D and L_D is the original number of terms in the same document D and other terms in the formulae are previously defined. However, such value is probably expected to be small in long documents since such ones will have higher number of terms but it might probably have an effect on shorter documents. As a simple example, consider a mixed document with a length of 93 words, 7 occurrences of the term 'inheritance' and a frequency of 9 times for its Arabic translation(s), which is ('الوراثة'), but in different positions of the document. Among these frequencies, the two terms co-occurred 4 times. With these simple statistics, formula 4.4 would result in reducing the number of words to 89.

4.2.2.4 Document Frequency Estimation

Document frequency estimation depends on how frequently a certain query term or one of its corresponding translations occur in all documents, regardless of their languages. Assuming that terms are synonyms across languages, it is reasonable to count every document that contains each of these terms in the DF statistics. Accordingly, if a document D includes at least one translation a_i, that document can be handled as if it contains the query term q and vice-versa. This would minimize the problem of biased document frequency because the document frequency will be computed across all documents (those in Arabic, English and mixed documents). Thus, if the source query term, for example, appears in many documents, whereas one of its translations occurs only few times (high weight), the result list will not be skewed towards that translation, as the document frequency will be computed as a joint document frequency containing all documents that include the source term or one of its translation(s). Formally, cross-lingual joint DF is computed as:

$$\grave{DF_q} = DF_q \cup_{\{i|i \in T_q\}} DF_i \quad (4.5)$$

Where DF_q is the set of documents which contain the source language term q in monolingual and mixed documents, $\grave{DF_q}$ is the new computed document frequency of the source term q in all documents in the collection regardless of the language(s) present in these documents (this is the joint DF of synonyms across languages), DF_i is the set of document which contain any translation a_i in documents, thus, the $DF_{T_q,D}$ is the set of documents in which one or more terms in the set T_q in the document collection occur and other terms are defined above.

If the Kwok formula (see equation 2.33), which alters the union operator (∪) to a normal summation operator (+) for simplicity, is used - as in this thesis, equation 5.5 would become:

$$\grave{DF}_q = DF_q + \sum_{\{i|i \in T_q\}} DF_i \quad (4.6)$$

Applying formulae 4.5 and 4.6 on the earlier example, Table 4.3 shows their joint DF(s) (the cross-lingual combined set of documents with each document contains one or more of the cross-lingual synonymous terms). It is notable that the max DF using the proposed cross-lingual variant of the Kwok's formula would be the total number of documents in a collection.

Approach	Synonyms (inheritance $_{\text{and}}$ الوراثة)
	Joint DF
Cross-lingual using Pirkola approximation	10
Cross-lingual using Kwok approximation	13

TABLE 4.3: The joint DF for the reference collection example.

If both joint TF and joint DF are applied in the reference corpus of the illustrative example in section 4.2.1, then results in Table 4.4 would be obtained as the new IDF will be log(14/10) = 0.14613.

Docs	Synonyms (inheritance $_{\text{and}}$ الوراثة)	
	TF * IDF	Scores
D_1	1 * 0.14613	0.14613
D_2	1 * 0.14613	0.14613
D_3	1 * 0.14613	0.14613
D_4	3 * 0.14613	0.43839
D_5	2 * 0.14613	0.29226
D_6	1 * 0.14613	0.14613
D_7	2 * 0.14613	0.29226
D_8	2 * 0.14613	0.29226
D_9	2 * 0.14613	0.29226
D_{10}	4 * 0.14613	0.58452

TABLE 4.4: Computations of ranking in the sample collection using both the joint TF and DF.

The rankings of documents are changed totally from those in Table 4.1. In particular, the rankings results in retrieving D_{10} followed by D_4 at the top while D_5, D_7, D_8 and D_9 earn the same scores starting from rank 3 in the list. At the final ranks documents, D_1, D_2, D_3 and D_6 are placed with the same scores. Apparently, these rankings are more reasonable, at least in terms of TF and IDF computations, than those presented in Table 4.1. This is because using cross-lingual structuring causes documents to be more comparable in that the term frequency and document frequency, which are the major components in ranked retrieval models, expand their computations to be cross-lingual and, thus, the source term and its candidate translation(s) are handled as instances and as if they are a single term presents in a single language.

4.2.3 Weighted Document Frequency or Inverse Document Frequency

It was described that traditional over-weighting often occurs when scores of documents in small collections are preferred because of using a single index for all documents, regardless of their languages. This always happens in multilingual collections, in which only several monolingual documents in different languages are used, or in both multilingual and mixed collections, in which mixed documents besides those monolingual are also indexed. As an example for how terms of documents in small collections can be favoured, assume that a mixed query ' مفهوم الـ deadlock' is posted to a multilingual Arabic-English collection with 70,000 documents in English and 6,000 documents in Arabic. After the query is bi-directionally translated and concatenated, it would result in approximately a mixed merged query like 'مفهوم الإقفال concept deadlock'. Using this query, the top ranked documents are probably expected to be dominated by those containing the terms مفهوم and الإقفال, rather than the terms concept and deadlock. This is because when all documents are placed together into a single collection, the N value (number of all documents) in the IDF factor of terms in the standard weighting scheme, which is computed as log(N/DF), for the Arabic sub-collection will increase significantly to 76,000, instead of 6,000 (the approximate increase is about 12.7 times), resulting in overweighted Arabic terms. The English collection will also increase but at a slower rate.

In the case of mixed and multilingual collections, however, the overweighting problem can get much worse as overweighting due to mixture of text also occurs. This is especially true even if the proposed cross-lingual structure model, which moderates the latter type of overweighting, is used. This is because when the process of cross-lingual synonymy takes place as described in cross-lingual structured model, the joint document frequency of the cross-lingual synonymous terms, which are mostly technical, would likely result in relatively small final weights for these cross-lingually structured terms, although they should have higher significance as they are the most significant words in the mixed merged query. For example, the document frequencies for the terms 'inheritance' and 'الوراثة' were 8 and 5, respectively, in the reference corpus of the illustrative example above. With cross-lingual structuring, the DF would become 10 or 13 using either cross-lingual Pirkola or cross-lingual Kwok variants, respectively – as was shown in Table 4.3.

Contrary to this trait, weights of non-technical terms in mixed merged queries would be kept as they should be and, thus, resulting in relatively higher weights, compared to joint document frequency of the cross-lingual technical terms. This makes such non-technical terms (concept and مفهوم in the previous example) skew the impact of the remaining technical terms in the final scores of documents. Note that such type of overweighting is different from traditional overweighting. While in traditional overweighting, the overweighted terms were مفهوم and الإقفال (mostly those terms in sub-collections with small numbers of documents), the overweighted terms in the second case were the terms: concept and مفهوم (mostly those are non-technical).

Due to these drawbacks, the second approach to handle the mixed-language problem, which can be applied either individually or in a combination with the cross-lingual structured query model, in a centralized index is to re-weight document frequency or inverse document frequency of query terms

(each of which are used as will described next). This is mainly to handle traditional overweighting problem and/or suppresses the impact of overweighted terms due to mixture of texts in mixed documents.

Broadly speaking, re-weighting of document frequencies or inverse document frequencies of terms in the proposed approach is performed using weighted factors computed from the statistics of the corresponding sub-collections of these terms. In other words, a weighted factor is computed firstly for each sub-collection, e.g. Arabic sub-collection, in the whole collection. Next, these weighted factors, which are in fact down scaling weights, will be incorporated in the document frequencies or inverse document frequencies of terms in mixed merged queries, as will be described next.

The weighted factors (reduction weights) of sub-collections can be computed into two different methods, these are called in this thesis as *damping weight factor* and *relative frequency factor*. Both of the two methods are described below. The damping weight factors are incorporated in document frequencies of terms, whereas relative frequencies of sub-collections are merged in inverse document frequencies. Thus, the methods are incorporated in weight computations in different ways.

Both of the methods can be used either when the document collection consists of several monolingual documents/corpora or with mixed and multilingual corpora, in which many mixed documents, beside those monolingual, are present.

4.2.3.1 Sub-Collection Damping and Weighted Document Frequency

It was shown that the boost values in terms weights are hinged on the total number of documents in each corresponding sub-collection. In other words, the increased quantities are varied, depending on sub-collection sizes and, hence, boosted values behave rather differently. Therefore, weighted factors for terms should also vary according to sub-collections in which these terms occur. This makes sense because the assumption that all sub-collections in a particular multilingual (or mixed and multilingual) collection are equally important, although their sizes are usually incomparable, is not fair. The Web is evident of such incomparable sizes. In reality, particular sub-collections may have minor impact while others come across with more discriminating effect, for example, scientific English sub-collection versus scientific Arabic sub-collection.

Thus, the major assumption behind proposed re-weighted document frequency (or inverse document frequency) is based on that a sub-collection with a higher number of documents is expected to be more useful and have more significance than another sub-collection with a small number of documents and, thus, terms belonging to significant sub-collections should have higher importance than terms belonging to less significant sub-collections. In that context, a sub-collection with a lower number of documents among all the presented sub-collections in the whole collection, the higher reduction weight (higher damping factor) for terms belong to it and the less it should contribute in the result list. Consequently, query terms in a sub-collection with small size would contribute less in documents scores.

To achieve this goal, two steps are needed. First, is to compute a damping weight for each sub-collection in the entire collection. These are the damping weight factors of sub-collections. Second is to incorporate these weights in terms of weights computations, specifically in document frequencies of terms.

Bearing this in mind, the damping weight factor of a sub-collection aims to mitigate boosted values by tuning document frequencies of terms in a way makes their inverse document frequencies, during subsequent weight computations, decrease.

Let (N) be the total number of documents in a multilingual collection presented in different languages (L_1, L_2…L_n) and placed in a single index. In this collection, the sizes of its constituent sub-collections are (N_1, N_2 …N_n) where ($N = N_1 + N_2 +…+ N_n$) and the information about the sizes is obtained during the indexing time. The damping weight factor of a given sub-collection is computed as the number of all documents in the entire multilingual collection, which is (N), over the size of a certain sub-collection. That is:

$$DW_i = \frac{N}{N_i} \tag{4.7}$$

Where DW_i is the damping weight of the sub-collection i and N_i is the total number of documents in the same sub-collection i. Obviously, the damping weight is the inverse of a probability. Given that the damping factor will be incorporated with DF of terms, for example the term t_i, the probability that the document D, in which the term t_i occurs, belongs to sub-collection i is given by (N_i/N) and, thus, the inverse of this probability would result in the damping weight of the sub-collection in which that term occurs.

Using formula 4.7, the damping factor grows as the number of documents in its sub-collection decreases. Recall the previous example above. The damping factor for the English sub-collection would be computed as (76,000/70,000), whereas it would be (76,000/6,000) for the Arabic sub-collection, resulting in a higher damping value (12.67) for the Arabic sub-collection than its peer (1.09) for English. Apparently, the damping factor is sensitive to number of documents across the several language-specific sub-collections. In a single monolingual collection, where ($N = N_1$), the damping value is exactly 1.

The same value 1 would also be obtained if the collection is multilingual and mixed (mixed documents besides several monolingual sub-collections) and the cross-lingual structuring (proposed cross-lingual SQM) is performed. This is because such cross-lingual structured terms cannot be attributed to a single language (the probability that the document in which the cross-lingual term occurs would be 1 as Pr(a) ∪ Pr(b) = Pr(a) + Pr(b)- Pr(a ∩ b)) and, thus, resulting in 1/1. For example, if the technical English term compiler and its translation المترجم are handled as two instances (cross-lingually structured), this means that these terms can occur in monolingual Arabic documents, monolingual English documents or mixed bilingual Arabic-English documents. In other words, the English term compiler can occur either in a monolingual English document or in a mixed Arabic-English document. Thus, the total number of documents for a cross-lingual structured term will consist of the sum of the total number of documents on each monolingual sub-collection that corresponds to a given language plus the total number of mixed documents. In such situations, it makes sense to assume equality between the numerator and the denominator in equation 4.7, resulting in a 1 value for the damping factor of technical terms. In spite of

this, the damping weight of non-technical terms in mixed and multilingual collection would still be less than optimal. This is because terms belong to sub-collection with small number of documents may still get benefit from the increase caused by the total number of mixed documents when all documents are placed together. Recall that an Arabic term may occur in either a monolingual Arabic document or a mixed document. For example, assume that an additional 9,000 mixed documents are placed with the earlier multilingual collection, in which 70,000 documents are in English and placed with 6,000 documents in Arabic, resulting in mixed and multilingual collection. In this collection Arabic terms will benefit from this increase in N value (9,000 documents), as mixed documents cannot be attributed to Arabic or English sub-collections, resulting in a damping factor equal to (85,000/15,000) instead of (85,000/6,000). In probabilistic view, the probability of the document in which the term occurs will increase from (6,000/85,000) to (15,000/58,000).

Now the next step is to integrate this damping factor in weight computations. Such combination can be performed with DF computations. Adaptation of DF using damping weight aims to tune the DF component in a way such that terms occurring in a sub-collection with small number of documents will be assigned lower inverse document frequencies (and thus lower weights and higher document frequencies). According to this assumption, the DF is composed with the damping factor as follows:

$$\acute{DF}_i = DF_i * DW_i = DF_i * \frac{N}{N_i} \quad (4.8)$$

Where $\acute{DF}_i$ is the new document frequency of the term i in the query. The term i appears in the sub-collection i. DF_i is the original document frequency of the term i. The impact of this formula varies according to whether the collection is multilingual only or multilingual and mixed.

In a multilingual collection, which consists of several monolingual sub-collections, formula 4.8 would reduce importance of all terms in the mixed merged query. In particular, the IDF values for the terms would be exactly equal to their inverse document frequencies if each sub-collection in indexed separately or as if a distributed architecture, which distributes documents according to their languages, is used for indexing documents. For example, given some Arabic term a_i with a document frequency of 2,000 in the earlier multilingual collection, in which 70,000 documents are in English and placed with 6,000 documents in Arabic. The IDF factor for this term using the composed version of both the damping weight and the modified DF as in formula 4.8 is approximately (log(3) = 0.4771), instead of (log(38) = 1.5798). This is the precise value for the *IDF* of the term a_i if the Arabic sub-collection is indexed separately. Hence, scores of documents could not be biased towards the Arabic term a_i, resulting in concealing the traditional overweighting problem, whereas at the same time the overweighting, regardless of its type, is probably moderated.

In a multilingual and mixed document collection with cross-lingual structuring for technical terms, formula 4.8 would result in that weights of such technical terms in mixed merged queries would likely be kept. For non-technical terms they would be reduced (as their IDF/importance will be reduced), but the terms would still be overweighted, as they will benefit from size of mixed-document sub-collection.

To conclude, when the proposed cross-lingual weighting is applied before applying formulae 4.7 and 4.8, this would result in that weights of non-technical terms in mixed and merged queries would likely be

reduced (as their IDF will be reduced), whereas weights of technical terms, which are cross-lingual structured, would be kept, as the DF of cross-lingual structured terms would be multiplied by 1 in formula 4.8.

4.2.3.2 Relative Frequency and Weighted IDF

Instead of the sub-collection damping factor, a relative frequency for each sub-collection can be also computed. The relative frequency of a given sub-collection is computed by dividing the number of documents in that sub-collection over the total number of documents in the whole multilingual (or multilingual and mixed) collection, that is:

$$RelativeF_i = \frac{N_i}{N} \tag{4.9}$$

Where $RelativeF_i$ is the relative frequency of the sub-collection number i and other symbols are defined above. The relative frequency is a probability, as was described above. Since this probability indicates the likelihood of the document, in which the term occurs, belongs to a particular sub-collection, it is natural to incorporate this into term weight computation. In particular, the relative frequencies of sub-collections are incorporated in the inverse document frequencies of terms.

To achieve this, the logarithms of the relative frequencies of sub-collections are firstly computed. Formally, the logarithm of the relative frequency, denoted below as FR_i, of a sub-collection i is computed as:

$$\begin{aligned} FR_i &= log(RelativeF_i) \\ &= log(\frac{N_i}{N}) \end{aligned} \tag{4.10}$$

The hypothesis of using this relative frequency along with its logarithm has two aspects. Firstly, the relative frequency measures the number of documents in a certain sub-collection with respect to the whole multilingual collection. This would map the number of documents of a certain sub-collection into the range from 0 to 1. Secondly, unless the value 1 is obtained as a result for the division operation, the logarithm of the relative frequency will always result in a decrease value. Accordingly, this would result in the following formula:

$$\begin{aligned} Weighedt\_IDF_i &= IDF_i + log\left(\frac{N_i}{N}\right) \\ &= log\left(\frac{N}{DF_i}\right) + log\left(\frac{N_i}{N}\right) \\ &= log\left(\frac{N_i}{DF_i}\right) \end{aligned} \tag{4.11}$$

Where $Weight\_IDF_i$ is the re-weighted IDF, which will be combined non-linearly with the TF of the same term i and IDF_i is original IDF of the term i. Recall the example above, in which the Arabic term a_i with a

document frequency of 2,000 is used. Using the tactic equation 4.11, the re-weighted IDF results in ($log(3) = 0.4771$). In that context, over-weighted terms, mostly non-technical in multilingual and mixed collection, can be assigned lower importance than those technical.

To this point, it is observed that the impact of formula 4.11 on the IDF of a particular term is similar of the effect of multiplying the original IDF, before applying the logarithm, with the relative frequency of its corresponding sub-collection in the entire multilingual collection, that is:

$$Weighted\_IDF_i = \log[\left(\frac{N}{DF_i}\right) * \left(\frac{N_i}{N}\right)] \quad (4.12)$$

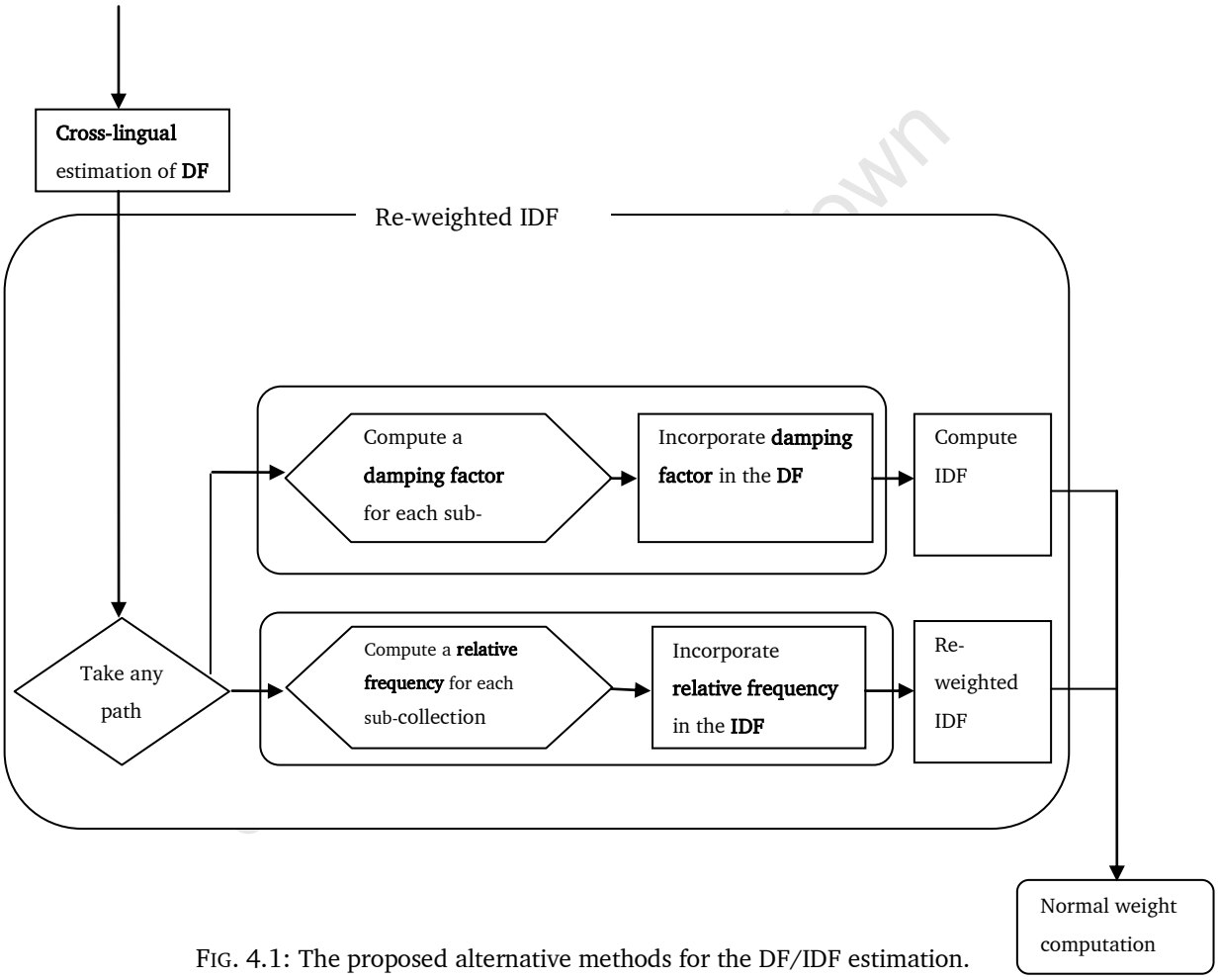


FIG. 4.1: The proposed alternative methods for the DF/IDF estimation.

Figure 4.1 illustrates a general flowchart for the proposed methods and their sequences of steps to re-weight IDF in mixed and multilingual collections whenever the cross-lingual structured model is used. However, the methods can be also employed in multilingual collections only.

4.2.4 Computing Document Scores

Once all the three essential components (term frequency, document frequency and document length) are estimated, the next step is to compute the scores of documents. Firstly, Table 4.5 summarizes the utilized

formulae, represented by their numberings, for estimating these components in a tabular form and the possible combinations among them:

Component / Options	TF	DF	Re-weighted IDF	Length (*L*)
Approach 1	(4.3)	(4.5) or (4.6) then (4.8)	-	(4.4)
Approach 2	(4.3)	(4.5) or (4.6)	(4.11)	(4.4)

TABLE 4.5: Possible combinations of the proposed formulae.

These estimated components are not linked to a particular IR model. Theoretically, the components can be used in any standard term weighting IR model function (tf * idf). However, experiments in this thesis were applied using the extended version of Okapi BM25 model to multiple weight fields (see section 2.1.3.2.2), after choosing one approach from Table 4.5.

4.2.5 Why not Use Translation Probabilities

Translation probabilities have been shown to be important in CLIR. For this purpose, research on CLIR proposed several algorithms that attempt to choose the translation candidates, whose probabilities are high, according to some resource, for examples, several dictionaries, unlinked corpus and parallel corpora. In spite of this and conversely to these approaches the proposed weighting algorithms, which were illustrated above, did not apply any translation probabilities mechanism. In particular, the proposed weighting assumes that each candidate translation for a source query term, particularly those are technical, is considered as being equally likely. However, one might ask why not use such approaches of translation probabilities in the proposed weighting. The answer stems from the differences between searching and retrieval in a technical/specialized domain against searching and retrieval in general-domain news stories.

As previously shown most currently available test collections and evaluation series have focused upon general-domain news stories. Accordingly, most CLIR techniques, including translation probabilities approaches, had been tested using such news-genre collections. These techniques provide innovative developments and contributions to significant issues in CLIR as described before in the CLIR review chapter. Even though, the likelihood of mismatching of these approaches may still occurs when migrating from news to more technical domains. In fact, some researchers (Rogati and Yang, 2004) confirmed that excellent performance in CLIR was noticed when utilizing news-genre collections. This is because the efficiency of MT systems, which usually employ news-based collections, has been improved by research throughout the decades. The remarkable effectiveness of performance is also obtained as a manual choice of high quality general domain training resources was made. For instance, Nie (2010) illustrated that the success of such techniques comes from the fact that the used corpora, such as the Canadian

HANSARD (see section 2.2.3.3), are clean input (Nie, 2010). The work of Rogati and Yang was one of the early studies that introduced these issues.

According to these observations, this may mean that current techniques are not set to be directly applied to scientific domains, especially in multilingual scientific non-English collections. In such cases of collections, there is a likelihood of poor retrieval because collections are often diverse in their sizes, genre, vocabulary and other domain-specific aspects. These aspects may vary solely from one domain to another (i.e. news-genre corpora vs. medical corpora). In fact, a significant diversity in collections quality is found from one domain to another, resulting in a possibility of discrepancies in the matching process when a certain approach, such as incorporation of translation probabilities, is tested in another domain. These conclusions were confirmed by Rogati and Yang, who compared different CLIR competitive approaches, which were proven to be robust when trained using news-genre, on test documents in the medical domain. In that work they found that the performance radically dropped when using the domain-specific training corpus.

One might take the news-genre as an example. Usually, news collections have unique characteristics that are not provided in other genres, such as computer science (Gey et al., 2005). Such characteristics include the use of general purpose vocabulary and the particular style of writing. In contrast, technical and scientific domains, as in which experiments in this thesis are to be applied, usually have rapidly developing terminology added to languages, especially in the non-English languages, e.g., Arabic. Furthermore, news genres usually employ little use of dialects as well as the regular and the wide use of proper nouns for places and names. Contrarily to the news genres, technical domains, especially in a large region like the Arabic-speaking world, have a diverse regional and synchronic terminology. News genre is primarily written in a single language. This is not the case in technical domains, especially in non-English languages.

With respect to techniques that incorporate translation probabilities approaches, mostly they employed newswire test collections. For instance, experiments reported on probabilistic structured query were carried out using TREC 2002, which is an Arabic newswire taken from the AFP. Employment of such newswire test collections makes such developed techniques do not consider the attributes of the target corpus and its domain while making these approaches use translation probabilities that are already unified to match the target corpus usage. For example, the words ‘object’ and ‘Oracle’ might have valid entries with different meanings/ alternatives, each of which can be probabilistically estimated in general-purpose dictionaries. However, the same words are very specific if the searched domain is in common computer science. Therefore, for the news domain it might be suitable to retrieve documents that contain the most probable translation in the target corpus, rather than including all of them, when a set of synonymous translations are present. However, this can be considered as an undesirable behaviour in technical jargon as this criterion of choosing the most probable translation does not hold. Particularly, the converse is quite accurate, especially for a language with several regional variations – as in the Arabic-speaking world, which was shown that it includes many countries, many of which have their own academy for the evolution of language (The Academy of Arabic Language, 2011). In such cases there may be a very highly relevant document that contains relatively infrequent translations, which is

pertaining to a specific region/dialect but it does not for others. Consider the Arabic translations for the technical English phrase 'object oriented programming' when the target corpus is in common vocabulary of computer science in Arabic. The translations are: 'البرمجة الشئية', ' البرمجة كائنية التوجه' , ' البرمجة موجهة الأهداف' and 'البرمجة كائنية المنحى'. All these alternative translations can be used in scientific Arabic documents, but according to the dialect/tongue of the writer. Technical topics in Arabic computer science domain exhibit this specific behavior. Thus, the appearance of what seems to be a superfluous translation like ' البرمجة الشئية' in documents does not make such a translation as an undesirable or irrelevant. Besides the very specific translations of technical terms along with the non-availability of suitable trained translation probabilities in computer science, these facts and observations lead us to assume that translations are equally likely in the proposed weighting method to accomplish the goal of retrieving many highly relevant documents on top, regardless of their regional variations.

4.3 Mixed-Languages in Separate Indices

As another option this thesis attempts also to propose solutions for the problem of information retrieval with mixed-language queries and documents when a traditional distributed architecture (see section 2.3.2) is used. However, it is important to confirm that the thesis did not target to solve the problem in terms of distributed information retrieval, as discussed earlier. Instead, it attempts to handle mixed-language phenomenon when documents are placed in separate indices and a single IR model for retrieval is used across them. This is what is known as the traditional distributed architecture.

4.3.1 Why a Distributed Index is not Optimal for Mixed-Languages

It was shown in CLIR review chapter that the dominant approach in distributed architectures is to translate a user query to target language(s) and next a monolingual language-specific search is carried out per each sub-collection followed by a merging method. In this context, distributed architectures provide users with only two alternative options to handle multilingualism in queries and documents.

The first option, as illustrated earlier in the introduction chapter, is to divide – even if implicitly using tools –information populated in each mixed document according to its languages across all/some of the language-specific sub-collections. As it was shown also, such an approach may result in loss of information depth or meaning in those mixed documents. This is an undesirable scenario in IR. Furthermore, mixed documents, even if they are highly relevant, would probably be *underweighted* in their corresponding sub-collections, as they are partitioned.

The second option, which was described also in see section 2.3.2, that can be applied to multilingual documents by the distributed approach is to index all documents, regardless of their languages, in a single unified big index. Next, each translated query is used to search against this single index. Afterwards, a merging process to obtain the final ranked list is applied.

At first look, this method sounds more adequate for multilingual documents because it circumvents partitioning mixed documents. But the approach has a drawback, as described earlier, that is the

overlapping of documents (documents are ranked on more than one individual list). This is especially true for the mixed documents, as they are written in more than one language. The assumption in IR studies when discovering such overlapped documents is that: since these documents appeared in more than one list, they are more likely to be relevant than those appearing on a single list and thus such documents should be promoted to higher ranks when individual lists are merged (Chen and Gey, 2004). One approach to apply such methodology is to sum up the scores of these overlapped documents. However, such an approach would result in a similar shortcoming when using the centralized index, that is the highly ranked documents at the top of the final retrieval list are expected to be mixed (because of the scores summation) rather than the highly relevant documents whether they are monolingual or mixed.

From these trends, a bridge can be seen between the user's information needs and relevant information when indexing mixed documents in a distributed architecture.

4.3.2 Hybrid approach of Indexing

Given a multilingual collection containing several monolingual document collections along with a mixed documents collection, in which each document is written in two languages, e.g., in both Arabic and English, a more appropriate architecture that aims to handle most problems which stem from the use of the two indexing architectures (distributed and centralized). In particular, problems like overweighting, partitioning and overlapping of mixed documents should be evaded.

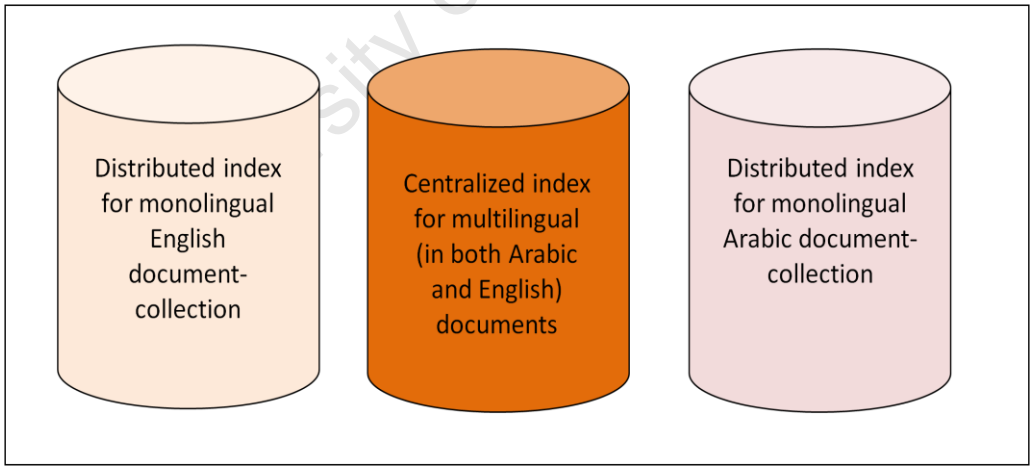


FIG. 4.2: The combined approach for MLIR.

Intuitively, a possible solution is to integrate both the centralized and the distributed approaches, taking advantage of their benefits, while attempting to minimize their drawbacks. Such a hybrid approach of both centralization and distribution mechanisms can help deal with both monolingual and multilingual documents. On one hand, a typical distributed architecture does not prefer collections with small number of documents, as in a centralized architecture. Accordingly, a more appropriate approach for indexing monolingual documents only would probably be provided by creating several monolingual distributed

indices, with one index for each monolingual language used in its corresponding sub-collection. In this way, multilingual documents would not be contained in these several distributed-monolingual-sub-collection(s). Such a single distributed index is called the '*distributed-monolingual-sub-collection'* – see Figure 4. 2.

Besides the overweighting avoidance, the significant benefit of indexing monolingual documents only in distributed indices is that the retrieval performance of each monolingual sub-collection run is expected to be efficient, due to the similarity in languages between documents and their corresponding translated queries, especially with only partial translations of mixed queries. Additionally, since mixed documents are not indexed here, the approach prevents partitioning them, and, hence valuable information placed in those documents will not be lost. Under the same context, the problem of overlapping of mixed documents across individual lists is also handled, since documents in all distributed-monolingual-sub-collection(s) are mutually exclusive.

On the other hand, a centralized architecture maintains indexing, searching and retrieval from a single index, regardless of the documents' languages and thus no merging process is required. This feature is very useful for multilingual documents because a retrieval process from a single index is expected to perform better than individual retrieval followed by a merging process. Therefore, a possible straightforward heuristic is to form a unique centralized index for multilingual documents only, but not for documents in several monolingual languages. Such type of index will be called the '*centralized-multilingual-sub-collection*'. Indexing mixed documents only in a centralized sub-collection index has major advantages. Firstly, it avoids partitioning these documents across several distributed indices. This is a major problem in the traditional distributed approach with respect to mixed documents, as illustrated above. Secondly, the use of the proposed approach under the assumed context makes relevant mixed documents more competitive to monolingual documents when results are merged because they are not partitioned and each of them is entirely retrieved with a full score.

Thirdly, the proposed approach minimizes the overweighting problem, which is inevitably in the centralized architecture, to its lowest level because monolingual documents are not included in this centralized-multilingual-sub-collection index. Thus, the number of documents in the entire multilingual collection (N) will not increase and consequently the IDF for a query term will be moderated.

Additionally, the use of this combined approach has also two additional advantages with respect to rankings. First, unlike the traditional centralized architecture, it breaks the dominance of mixed documents at the top of the retrieved list. Secondly, unlike the traditional distributed architecture, the proposed combined approach makes mixed documents competitive to those monolingual ones. This is because each sub-collection in the combined index will include many relevant documents in the final merged list - depending on the used merging methods. In fact, if an individual monolingual/mixed ranked list in each sub collection (distributed-monolingual or centralized-multilingual*)* contains many relevant documents in the top rankings, then these documents would likely be included in the final ranked list.

Despite the use of the centralized index, the entire architecture of the proposed strategy can be considered to be also distributed. The index can be viewed as a big combined repository for indexing a

centralized sub-collection that is inserted on the same level with other distributed (monolingual) sub-collections.

Since the goal is to focus on broadly applicable techniques for handling mixed documents and queries, one should note that the proposed indexing architecture can be easily adapted to other languages, not only for the Arabic and English pair, making it a practical solution especially in technical domains. However, in realistic collections environment, it is possible to have mixed documents in several bilingual-languages, for examples, Arabic and English or Chinese and English. If the multilingual and mixed collection contains such cases, it would be appropriate, beside the several monolingual sub-collections, to create several centralized-multilingual-sub-collection(s) also, with one sub-collection (centralized index) for each bilingual languages. For instance, if there are several monolingual documents in Arabic, English and Chinese that are placed with several mixed documents in both Arabic and English languages along with some mixed documents that are written in Chinese and English, then five indices will be produced using the proposed architecture of indexing: two centralized-multilingual-sub-collections for each bilingual language pair and three distributed-monolingual-sub-collections, one for the Arabic language, the second for the Chinese and the third for the English language. Due to such kind of utilization, it is possible to say that the proposed solution of indexing is scalable and flexible.

However, although the proposed indexing approach mitigates the overweighting problem (in particular, the monolingual overweighting) along with the beneficial features that were illustrated above, but it still appears, meaning overweighting, due to mixture of texts which causes biased TF and biased DF problems - as documents in the *centralized-multilingual-sub-collection* are mixed. Overweighting caused by text mixture, as described earlier in the introductory chapter, can skew the result list towards terms with low DF, even if their translations have higher document frequencies. Furthermore, weighting of similar terms across languages is still independent. In addition, problems like neighbouring pairs in two languages and their effects on TF are still present.

To overcome these problems, it is possible to utilize the same proposed approach of the cross-lingual structured query model in section 4.2.2. But, another different method for structuring terms cross-lingually is pursued. In particular, the methods, which called *probabilistic cross-lingual structured query model*, is different only in that the decaying factor (see section 4.2.2.2) of the co-occurring terms in different languages is estimated in a probabilistic nature.

Nevertheless, one might ask: what is the benefit that could be gained from using such a probabilistic approach since a simple frequency-based counting, as applied previously in the cross-lingual structured model, suffices?

Besides exploring other ways, it is all about reasoning. Generally, probabilistic approaches claim benefits based on probability theory. Grossman and Frieder (2004) stated that non-probabilistic techniques may have a particular arbitrary feature. Although they may practically perform well, they may also lack a concrete theoretical notion due to the difficulty of estimating parameters unless a sufficient training data is employed. Otherwise the model may result in an inaccurate estimation. Grossman and Frieder showed also other examples like the difference between the probabilistic model in IR and the other models. The authors, for example, present the classical debate on relational to object-oriented database management

systems. According to the authors, the latter lacks sound theoretical basis whereas, the relational model does.

Furthermore, IR for example, as stated by Gao, et al. (2006), can be viewed as a reasoning process that attempts to decide whether there is a relationship between queries and documents and whether this relationship is strong. Accordingly, the process may involve any type of reasoning, including statistical reasoning. According to Gao, the popularity of statistical reasoning in IR is due to the nature of the available knowledge in this field, in which much of studies are of statistical nature. Furthermore, the studies often incorporate probabilities in the different components of IR for different purposes, for examples, probabilistic structured query for TF and DF estimation, statistical translation disambiguation, etc. Thus, estimation of TF, as an example, based on both its individual occurrences and its co-occurrence with its translations in mixed documents in CLIR is not an exception. Furthermore, statistical framework assists researchers to conclude accurate estimation and provide evidence and basis for future integration of similar statistical approach in IR.

To this end, it should be noted that the proposed probabilistic-based and frequency-based counting approaches are not paradoxical and also their results, but they can be recognized as two different version approaches.

4.3.3 Probabilistic Cross-lingual Structured Query Model

The proposed approach, which is called *probabilistic cross-lingual structured query model*, is based on reducing the effect of co-occurring terms in doubling the weights. Given that the source term is *a* and its translation is *e*, the aim is to suppress the TF of *a* or *e* but not both since the process of reducing the TF of synonymous terms across languages is logically not symmetric, meaning that the TF of co-occurring terms should be added either with TF of *a* or with that of *e* – as was shown in section 4.2.2.

To achieve the above-stated goal, the probability of individual/exclusive occurrence of the source language query term *e*, excluding those *a* terms co-occurred with *e* is to be firstly computed. This probability will be called *diminishing probability*. Secondly, the diminishing probability is then incorporated in the TF of the synonyms across languages and in the document length computation. The details of the approach are illustrated below.

4.3.3.1 Diminishing Probability

Assume that in a document d_k with a length L_k, the absolute frequencies of terms *e* and its translation *a* are tf_e and tf_a, respectively. In the same document, the number of times in which the term *a* co-occurs with the term *e* is tf_{ae}. The document has a uniform distribution on its words. To find the number of occurrences of the term *e* alone – without its co-occurrence with the term *a*, Pr ($e \cap \bar{a}$) is to be computed. Pr ($e \cap \bar{a}$) is the probability that term (e) occurs in document (d_k) but without co-occurring with *a*.

But, given the information that the term e has also occurred in the document, Pr ($e \cap \bar{a}$) is to be quantified. This is the conditional probability of $\bar{a}$ given that e has occurred, denoted as Pr ($\bar{a}|e$). Note that when re-assessing the sample space to only e and a, $\bar{a}$ is exactly the exclusive/independent occurrence of e. Similarly, if the individual occurrence of a is to be computed, then the probability Pr ($\bar{e}|a$) is to be estimated.

However, to compute the conditional probability Pr ($\bar{a}|e$), its complementary probability Pr ($a|e$) is to be computed, firstly. Pr ($a|e$) is the probability of the occurrence of a, given that the term e has occurred. From statistical point of view, the conditional probability Pr($a|e$) is defined as:

$$\Pr(a|e) = \frac{\Pr(a \cap e)}{\Pr(e)} \tag{4.13}$$

Whereas, the complementary probability Pr ($\bar{a}|e$) of Pr ($a|e$) is:

$$\Pr(\bar{a}|e) = 1 - \Pr(a|e) \tag{4.14}$$

In that way the diminishing probability, Pr ($\bar{a}|e$), can be obtained. Next this diminishing probability will be incorporated when estimating TF. Clearly, the scheme above corresponds to the case that only one translation is present in a mixed document. But, in Arabic technical documents, it is common to find more than one translation in the same document. Appearance of such phenomena increases the likelihood of the diminishing factor. In such case of n alternative translations, a possible solution is to consider all of them as synonyms, but in one monolingual language, firstly. The impact of such task will result in creating a set that contains all these translations, which in turn, will be considered as a single translation. Following this scenario allows applying the diminishing probability directly in equations 4.13 and 4.14.

An alternative approach is to find the union (∪) of all translations, rather than considering them as monolingual synonyms. For example, assume that two translations (a_1, a_2) are known for the term e. In such cases, the Pr ($a_1 \cup a_2|e$) is firstly computed, according to equation 4.13. Next, the decaying probability Pr ($\overline{a_1 \cup a_2}|e$) is to be computed. From a statistical point of view:

$$\Pr(a_1 \cup a_2|e) = \Pr(a_1|e) + \Pr(a_2|e) \tag{4.15}$$

And by generalizing this definition, the following equation will be obtained:

$$\Pr(a_1 \cup a_2 \ldots \cup a_n|e) = \Pr(a_1|e) + \Pr(a_2|e) + \ldots + \Pr(a_n|e) \tag{4.16}$$

Accordingly, the diminishing probability is the complementary probability, that is:

$$\Pr(\overline{a_1 \cup a_2 \ldots \cup a_n}|e) = 1 - \Pr(a_1 \cup a_2 \ldots \cup a_n|e) \tag{4.17}$$

To this point, assume that the absolute frequency of the term e is 10 ($tf_e = 10$). Two candidate translations a_1 and a_2 occur 7 and 8 times, respectively, in a document with a total number of tokens equals to 100, $L_k = 100$. The total number of co-occurrences of both e and a_1 together is 4, whereas the pair e and a_2 appears together 3 times. Using these assumptions, $\Pr(a_1 \cup a_2|e)$ can be predicted by applying formula 4.15 or 4.16 directly, resulting in:

$$\Pr(a_1 \cup a_2|e) = \frac{\Pr(a_1 \cap e)}{\Pr(e)} + \frac{\Pr(a_2 \cap e)}{\Pr(e)} = \frac{4}{10} + \frac{3}{10} = \frac{7}{10}$$

Accordingly, $\Pr(\overline{a_1 \cup a_2}|e)$ can be computed as:

$$\Pr(\overline{a_1 \cup a_2}|e) = 1 - \Pr(a_1 \cup a_2|e) = 1 - \frac{7}{10} = \frac{3}{10}$$

Which is the diminishing probability.

4.3.3.2 Incorporating the Diminishing Probability in TF

Since the absolute frequency of the term e includes also its co-occurrences with its translation(s), it is possible to incorporate the estimated diminishing probability in TF of that term. This is acceptable, as $\Pr(\overline{a_1 \cup a_2}|e)$ is the likelihood that term e occurs in the document independently. Such integration results in cutting-off the frequency of co-occurrences of the term with its translations while at the same time retaining the exclusive frequency of the term e. Hence, it suppresses the contribution of the term e to the final weight of the synonymous terms across languages. Formally, incorporating the diminishing probability with the absolute TF of the source language query term is computed as follows:

$$TF'_{q,D_k} = TF_{q,D_k} * Pr(\overline{a_1 U\, a_2 \ldots U\, a_n}|q) \quad (4.18)$$

Where D_k is a mixed document that is written in two language, TF_{q,D_k} is the absolute frequency of occurrence of the source language term q in document D_k, regardless its occurring exclusively or not, a_i is a synonymous translation for the term q, TF'_{q,D_k} is the new computed term frequency of the query term q and n is the number of the alternative translations that appear in the document For instance, when formula 4.18 is applied to the earlier example, in which $\Pr(\overline{a_1 \cup a_2}|e) = 3/10$, then the individual occurrence of the term e is exactly 3, instead of 10.

4.3.3.3 Estimating TF of cross-lingual Synonyms

Obviously, whenever computing the TF of synonyms across languages, it is important to note that reducing the frequency of the co-occurred pairs is not symmetric, meaning that the pairs' frequencies are either to be counted in the TF of the source language query term q or in the TF of its translation(s), but not in both.

The latter is computationally costly because formula 4.18 would be repeatedly applied to each pair containing the query term and one of each translation. Thus applying the same formula to the source query term side only is much easier. Accordingly, the TF of synonyms across languages can be computed

as:

$$TF'_{q,D_k} = TF_{q,D_k} * Pr(\overline{a_1 U\ a_2 \ldots U\ a_n}|q) + \sum_{\{i|i \in T_q\}} TF_{i,D_k} \quad (4.19)$$

Where all terms are defined above.

4.3.3.4 Estimating Document Length

Frequency reduction usually impacts the number of words in documents. In particular, the number of terms would be reduced by the frequency of co-occurrences of the source query term and one of its translation(s) – as it was shown in the proposed cross-lingual structured query model. However, before applying this reduction, assume firstly that the occurrence of any word in the document (d_k) is mutually exclusive. This would result in an equally likely probability of each word occurrence. Thereby, the probability that the term e occurs with its candidate translations a_1 and a_2 is Pr((e ∩ a_1) ∪ (e ∩ a_2)). Thus, the probability of occurrences of all other terms without this probability (of the co-occurring terms), denoted by Pr (t) would be:

$$pr\,(t) = 1 - pr\,(\,(e \cap a_1) \cup (e \cap a_2) \ldots \cup (e \cap a_2\,)\,) \quad (4.20)$$

Where all terms were defined above. With this definition, it possible to reduce the number of words in the document (d_k) using the following ad-hoc formula:

$$L'_k = L_k * Pr(t) \quad (4.21)$$

4.3.3.5 Estimating Document Frequency

The diminishing probability does not have any effect on the document frequency. This is clear from the major premise made about computing this probability. That is, the diminishing probability is computed for co-occurred pairs. However, the *DF* in the centralized-multilingual-sub-collection should probably be estimated because there may exist a mixed document that contains only the translation(s) of the source query term. Such document will be counted in the DF of its source term. Accordingly, the DF in the centralized-multilingual-sub-collection will be estimated as in equations 4.5 or 4.6.

4.3.4 Computing Document Scores

As in the proposed approaches within the centralized architecture, the proposed approaches in this section can be used in any IR model that weights terms according to standard function of term weighting (*tf * idf*). With respect to experiments of the proposed solutions in the distributed architecture, the same extended version of Okapi BM25 model to multiple weight fields (see section 2.1.3.2.2 and equation 2.23) was used to weight documents in all sub-collections. This is implemented after estimating TF, DF and document length components as illustrated above. However, documents were indexed using the proposed hybrid architecture of indexing, as described in section 4.3.2.

4.3.5 Retrieval and Result Merging

Thus far, results are retrieved individually from each sub-collection. In particular, the monolingual Arabic queries will be used to retrieve Arabic documents from the Arabic distributed-monolingual-sub-collection. The English ranked lists will be obtained by running the translated monolingual English queries against the English distributed-monolingual-sub-collection. For the multilingual-centralized-sub-collection, both translated Arabic and English queries are concatenated, as in the normal centralized architecture, to form a big query. Weights are modified in the index as in the probabilistic structured query model.

The intuitive next step is to merge all these individual ranked lists into a single result. Different merging methods were used in the experiments in this thesis, most them were described in section 2.3.2. The first used method for merging was the raw-score merging, which sorts all individual results by their original similarity scores that are obtained from each sub-collection. The second used approach for merging the results is the normalized-score merging, which adjusts/normalizes scores in each individual list before merging by the maximum obtained score in that list. The third approach also normalizes scores according to equation 2. 40. The fourth employed approach was the weighted score merging approach, which is based on both document scores and collection scores. The collection scores are based on the CORI approach (see section 2.3.2) and the computed scores of documents wer normalized according to equation 2.39. Figure 4.3 illustrates the components that were used in the proposed solution.

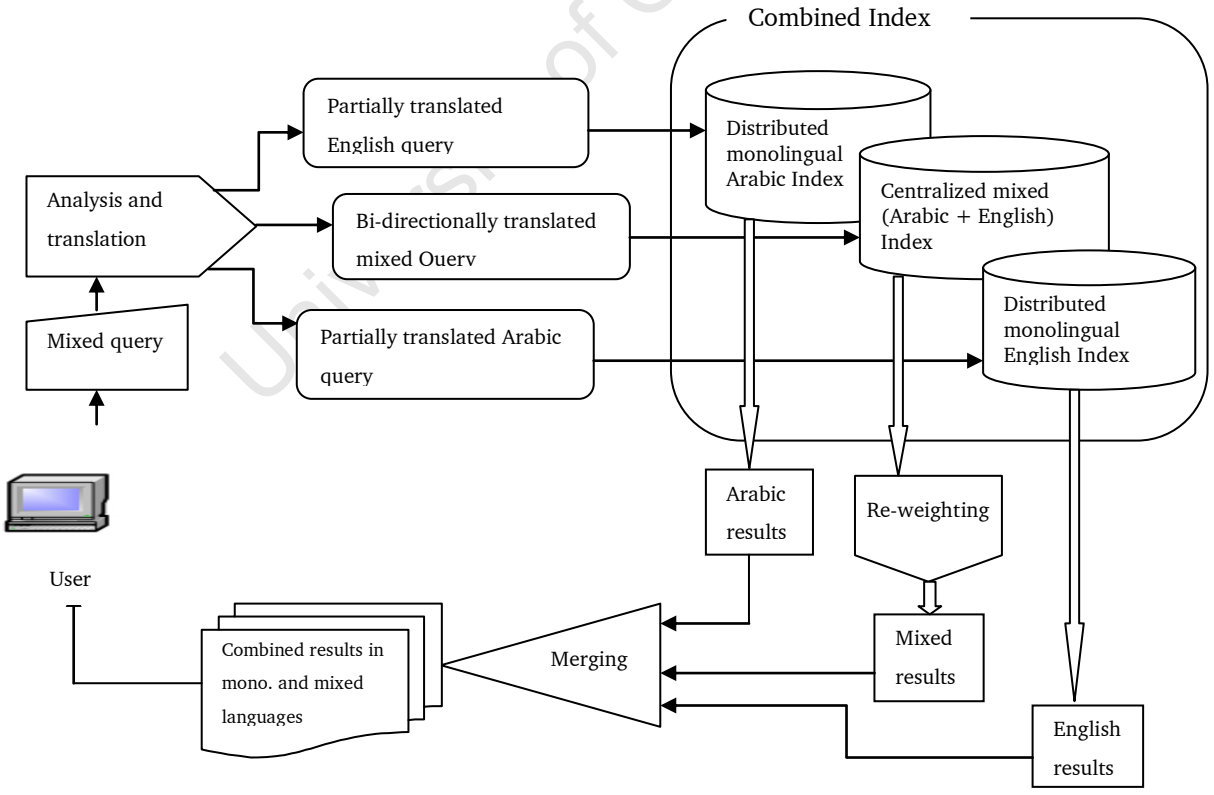


FIG. 4.3: The major components of the proposed solutions using separate indices.

4.4 Summary

In this chapter, the matching problem in information retrieval with mixed-language queries and documents is introduced. It is shown that current indexing and weighting approaches are optimized for monolingual retrieval and documents, rather than mixed documents with several term/snippets in different languages. The manifest issues in such document ranking weights are the multilingualism feature and the co-occurrences of bilingual pair terms in different languages. In particular, the latter phenomenon usually makes weighting functions skewed to mixed documents, resulting in adding extra weights for such documents. Therefore, within the structure of the major approaches of indexing (centralized and distributed), several approaches were proposed for handling the weighting components (TF, DF and length) in mixed documents. The proposed models describe novel techniques for alleviating this mixed-language matching as well as problems like overweighting, skewed TF, biased DF, etc. Additionally, a new re-weighting DF or IDF method was proposed. This is to handle overweighting problem within multilingual collections and mixed and multilingual collections as well. The weighted IDF can be used in isolation or in a combination with the proposed cross-lingual structured query model. Furthermore, a new hybrid approach for indexing documents in multilingual and mixed collections when a distributed architecture is utilized, is experimentally developed. This is because current approaches to indexing in MLIR also ignore the mixed-language feature in mixed documents. Additionally, a probabilistic approach for weighting mixed documents in the proposed hybrid indexing approach is also proposed. During the chapter, however, it is shown why the proposed models are language-aware solutions.

One motivation for this research is to develop a new test collection with mixed-language queries and documents. It was shown that most current test collections are built from news genre and the majority of them were monolingual. Therefore, the created corpus is on common computer science vocabulary, synchronic and multilingual and mixed in both Arabic and English. The next chapter describes how the created mixed and multilingual collection is gathered and how it was statistically tested.

CHAPTER

5

Building the Test Collection

Gathering a large amount of sample text/speech in a certain natural language, known as the corpus, is a common task in several research fields. For example, in the linguistics field, building corpora is a widespread activity when constructing dictionaries and thesauri, analyzing linguistic comparisons among languages, studying language syntax, acquisition and lexicography as well as examining many other pedagogical issues and related disciplines.

In text retrieval research, including CLIR, textual corpora in forms of documents, are being also used intensively for various purposes, as illustrated in review chapter. Examples include training translation models, extracting collocation and word co-occurrence statistics or devising algorithms for the different tasks of IR such as stemming, corpora alignment, etc.

Corpora are also used in IR to measure which technique is better than other techniques in a standard way. In such situations, corpora are referred to as test collections, as discussed in the literature. Test collection often consists of three types of sets: a set of documents, a set of queries and a set of relevance judgments for each query in the query set. But, since IR is a wide area of research, test collections in this field have evolved over the years to adapt the various changes in search application including data and user requirements.

For purpose of studying mixed-language queries and documents, as well as devising new algorithms that match this multilingualism phenomenon, a mixed and multilingual test collection is required. It is true that many ad-hoc text collections were developed, but most of them are not appropriate to conduct experiments reported in this thesis. Most currently available ad-hoc test collections, and almost all CLIR collections, are monolingual, rather than mixed. Multilingualism in both queries and documents can make nuance differences in developed algorithms and ranking functions. This is because developed approaches on monolingual test documents are not necessarily hold for multilingual and mixed test collections. Additionally, existing test collections are primarily focused upon general-domain news

stories. Although, such corpora had proven their significance in several fields but, yet this does not mean that they can serve as training corpora, e.g. training SMT, in other genres, such as those specialized and scientific. There is always likelihood for poor handling and performance. It is true that many domain-specific and specialized corpora were built (i.e. some Arabic collections, NTCIR and CLEF) but, most of them are not synchronic and collected from a particular country, for example in some NTCIR collections the majority of the collected scientific documents were compiled from Japanese or they are designed to work on bibliographic records that contain only titles and abstracts, rather than complete text documents. Furthermore, scientific collections cover only a few languages. Arabic is rare among them. The scarcity of specialized Arabic test collections is not new and it is evident if one investigates the currently available collections. This conclusion was also confirmed by many researchers (i.e. Saad and Ashour, 2010).

Therefore, a primarily Web-based multilingual and mixed Arabic-English test collection on common computer science vocabulary, with approximately 42 million words, has been created to serve as a benchmark for experiments reported in this thesis.

This chapter describes how this created test collection had being collected, processed, cleaned and filtered. The chapter discusses also, how the test collection was evaluated and validated using statistical tests. Hence, measures like distribution of words and vocabulary growth are illustrated. Furthermore, the chapter also describes how both the query set and the relevance judgments were constructed.

The rest of this chapter is organized as follows. Section 5.1 describes the common features of the created corpus. Section 5.2 is devoted to the created test collection. Firstly, it shows how the data set was collected, processed and assessed, along with the corpus statistics and the statistical tests that had been applied to validate the corpus. Observations about the corpus are also provided in terms of comparisons with the standard test collections. Secondly, the section presents also how the mixed query set was constructed, along with some observed characteristics and statistics on these mixed queries. Furthermore, section 5.2 illustrates how the query set is replaced with topic files. Thirdly, the section shows how the relevance of documents against queries were judged. Finally the summary of the chapter is provided in section 5.3.

5.1 The MULMIXEAC Test Collection: Common Features

Since building a test collection is a hard and a time-consuming process that requires a significant amount of manual work, the created test collection, which has been named MULMIXEAC (MULtilingual and MIXed English Arabic Corpus), was gathered primarily from the Web in the second half of the year 2011. Employing the Web as a resource makes it possible to collect large amount of data in relatively short time with cheap and limited resources, e.g., a Web crawler and a suitable server machine. In addition, the Web is free, diverse in both languages and contents and directly accessible from anywhere. These are strong features of the Web. Another useful characteristic of building a Web-based test collection is the inclusion of modern and up-to-date vocabulary. This is another important characteristic in specialized and technical domains like technology and computer science.

One might ask why gathering a specialized collection on computer science domain. It was previously discussed in section 4.2.5 that news collections have distinguished features that do not hold for other genres, especially those scientific. For instance, it was discussed that multilingualism (the use of more than one language in a single document) is common in the latter domain in non-English languages. This is caused by the fact that most terminology is borrowed from English. Furthermore, in news domains the regular use of places and proper nouns is widespread as discussed before in section 4.2.5 and the target readers, however, are mostly natives with different educational background. They may also be illiterate and thus, the writing style of the language is simple and does not need to be clarified in English as in scientific domains. Furthermore, it is common in news genre that the use of different dialects is not widely spread. This feature especially holds for Arabic, which has many regional variants. For instance, if one listens to the news in Arabic at Al-Jazeera or BBC, it is definitely noticeable that the same vocabulary is used by the both broadcast corporations. These differences between news collection and sciehtific specialized domain are major as they could affect the different phases of the IR process such as stemming, stopword removal, indexing, POS tagging, plural of words, especially in Arabic, etc.

With respect to standard test collections, it is notable that the majority of them and almost specialized CLIR collections are either written in a single language or consist of several monolingual documents/collections in different languages (multilingual), as in many TREC, NTCIR and CLEF campaigns documents or parallel as in the Springer collection (discussed in section 2.4.1.2) and some parts of the NTCIR-1 and NTCIR-2 (both collections contain English-Japanese paired abstracts). In particular, the majority of documents in these collections are written in a certain language, rather than documents with different terms/portions/snippets/ phrases/paragraphs that are tightly–integrated in multilingual and mixed forms. In spite of this monolingualism dominance, there are many mixed documents in some East Asian collections, but still their query sets are primarily monolingual, rather than mixed too, even if they are translated in another target language for CLIR tasks. Such underlying assumption of monolingual queries only for testing purposes ignores the great demands of multilingualism and multi-culture attitude, especially with their real presence in everyday life in regions like the Arabic world and East Asian countries.

Among the several scientific domains, the created test collection has been collected from texts in computer science genre. This is mainly due to the reason stated above, which is the multilingualism and mixing of texts. It is true that many scientific Arabic documents are mixed but in domains like computer science the vocabulary is emergent and every day new terms are added to the language, unlike other domains. Since the process of translation and/or translation is not performed on a regular basis, as discussed in the introductory chapter, the vocabulary in Arabic computer science domain is very different. This makes the selection of the computer science domain for building the test collection is optimal and a good environment for testing experiments in this thesis. Furthermore, the use of computer science vocabulary enlarges the chances for developing other techniques as the vocabulary would be well understood by IR researchers, unlike vocabulary in medicine, which needs extra efforts from the researchers in computer science.

Since the texts of the compiled collection were sampled from various categories of computer science, the document collection would appropriately be classified as a specialized collection. The languages of the texts that are presented in the test collection are Arabic and English, as this thesis focuses on these two languages. Thus, MULMIXEAC can be classified as a multilingual collection. However, it is also mixed because it contains a considerable number of bilingual Arabic-English texts. The query set of the collection consists of mixed-language queries along with relevance judgment files containing parallel texts. The document collection also contains documents written by Arabic writers from different Arabic countries and, thus, it is synchronic.

5.2 Building the MULMIXEAC Test Collection

The test collection contains three major components: a written text collection consisting of documents (data set); a set of mixed queries (query set) describing information needs and placed into topic files; and a set of relevance judgments, for a pre-defined search task, per each query in the query set and placed in parallel English-Arabic files.

5.2.1 Data Set

There are three well-known approaches to gather text corpora from the Web. These are: automatic crawling and harvesting based on a pre-defined list of URLs; automatic and/or manual downloading based on manual and/or automatic submission of queries to search engines; and manual collection of documents. With respect to the data set in the MULMIXEAC collection, all three approaches were utilized. The primary reason for this variety of approaches was caused by the fact that computer science documents in different languages are not always available, with adequate mass and varied contents placed into a suitable electronic format, to collect via crawlers. On one hand, English scientific content on the Web is rich and thus automatic crawling (first approach) can be used to download documents. Therefore, the first approach to collecting documents for MULMIXEAC was mainly used to download English documents. On the other hand, although there are many rich Arabic documents, digital Arabic content on the Web shows a lack of quality, especially in scientific fields, for examples physics and technology. In many cases Arabic scientific information may be placed in references, articles, essays or books that are strictly accessible on the Web or in electronic media owned by institutions and universities. Hence, in order to acquire such controlled resources, along with the need to extract suitable Arabic documents from the Web, both the third approach (manual collection) and the second approach (manual collection based on queries) to gathering data was used to collect Arabic content (mixed documents and monolingual Arabic documents). However, this does not mean that the first method to data collection, which is automatic crawling, was not used to collect Arabic documents. In fact, many Arabic documents were collected also using the first method.

Nevertheless, the overwhelming majority of the collected documents were mainly produced through both the first and the third approaches (automatic crawling and manual collection, respectively).

5.2.1.1 Data Set Collection

In the first approach, which is the automatic crawling, a list of URLs was firstly prepared by a group of 25 scholars, primarily teaching staff, in the field of computer science at two Arabic universities. Each member in the group was asked to create an initial set of URLs. After being produced and collected from participants, all the sets were pooled. Repeated URLs were discarded, and finally a cleaned list of URLs was obtained.

There is another method that could be followed here to create the URLs set. It was applied by Sharoff (2006) and Baroni, et al. (2006). In that method, random queries created from the most frequent words, in terms of collection genre and target, are first generated. Next, these queries can be posted to a certain search engine, e.g., Google, using its search APIs. Following this, top *n* hit pages per query are kept as URLs. URLs that were obtained by all queries then serve as seeds to the Web crawler after stripping all duplicates. However, using such an approach is mainly based on obtaining a search engine API key, which is strictly not workable at some Arabic countries due to political considerations.

Turing the attention to documents collection, the authors and copyrights owners of each URL were contacted to obtain permissions for contents dissemination. Unfortunately, the response was weak. This is not an easy challenge because it probably means that the contents of the created test collection cannot be re-distributed. Meanwhile, the test collection would be only available in the Digital Libraries laboratory at the Department of Computer Science, University of Cape Town. It would also be available for those who participate in its construction.

Following this step, the produced list of the URLs was utilized for seeding the *WebReaper* Web crawler [15]. WebReaper has the ability to download pages at a given main URL and then it follows a recursive process in traversing and downloading other linked pages. Thus, WebReaper was used to make local copies of documents from the specified URLs (seeds) in the list without any criterion on the type of the harvested document, e.g., html, Pdf, doc, etc. The process was run intermittently, rather than continuously, and from time to time during the second half of the year 2011.

A manual collection of data was also considered. In particular, a group of 100 students/tutors, who are Arabic native speakers, at different academic levels at some Arabic universities were asked to collect documents on common computer science topics. The members of the group primarily employed the second and/or the third approaches to data collection, which are the manual submission of queries to search engines and the manual collection, respectively. Some students submitted their own random queries to some search engines and then downloaded the retrieved pages manually. Others downloaded documents from specific websites, which are popular to them. A considerable number of students also extracted documents from their academic homework reports, academic essays and articles and from some graduation projects (in the case of finalist students). Note that the produced texts using the latter approach (extraction from essays and articles) were primarily mined from documents that were written as a medium to inform about a topic is specific language and not to practice the language(s) itself. This is

[15] http://www.webreaper.net/

important because it results in making the MULMIXEAC a partially learner collection. A learner corpus (McEnery et. al, 2006) is a corpus written in the second language of native speakers or learners of foreign languages, whose native language is different from the language of the corpus. Moreover, learner corpora help a lot in understanding the pedagogical aspects in order to improve grammatical and lexical needs. Furthermore, they show, for linguistics, whether there is the writing style of learners is deviated from the standard vocabulary of a particular language (Pravec, 2002).

Contents of the document collection are mainly collected from references, papers, websites, books, online help, students essays and articles, software documentation, forums, patents, etc, but yet all of them are from computer science. All the collected pages and documents, regardless of approach used for gathering, were merged together into a single pool. This results in a total size of 14.2 GB of raw text data with a total number of documents equals 90,583. Table 5.1 illustrates number of documents in each language along with those mixed. The sizes of the languages (Arabic, English and mixed texts) are not similar because it is subject to the availability of adequate texts in each. In particular, the size of the English documents is the highest one, whereas mixed-language documents represent approximately about 30% of the total number of documents in the raw collection.

Description	Language(s) details	Numbers	%	Total
Number of documents	Monolingual English	62,753	69.28	90,583
	Monolingual Arabic	1734	1.91	
	Multilingual(mixed Arabic and English)	26,096	28.81	
Corpus Size (GB)		14.2		

TABLE 5.1: Statistics of the MULMIXEAC collection. Figures are provided before cleaning the corpus.

5.2.1.2 Collection Processing

After gathering the collection, it was processed in order to create a cleaned collection in HTML format. The cleaning process was conducted into two phases. The first phase resulted in a roughly cleaned collection, named the MULMIXEAC version (A) in which figures, HTML tags, etc are preserved. In the second phase of cleaning the document collection, a pure textual collection, named as the MULMIXEAC version (B), was extracted. This was the employed version for the indexing stage.

5.2.1.2.1 First Pass Cleaning

In the first pass of processing, pages and documents with trivial sizes, which were found relatively frequent, were removed. Following this, documents were also manually explored so as to find out what types of tools and/or application codes would be used and/or developed. Documents were found in several different formats (SHTML, HTML, DOC, TXT, RTF, PDF, etc). Accordingly, on the first step of

cleaning, pages in different HTML formats were automatically processed, while preserving the same format of files. This primarily includes the removal of weird symbols (®, §, ™), fixed comments and navigational data. Throughout this cleaning, formulae, ellipses, figures, mathematical symbols, images, HTML tags and punctuations were not discarded during this phase of creating version (A) of the document collection.

Following this, documents that were created by word processor software, were processed, e.g., those in formats like doc, RTF, txt, etc. In particular, an application was developed so as to create HTML files from such files. Furthermore, some applications (i.e. Microsoft word), which have the ability to create HTML files from a file that is saved in their formats, were also used for this purpose.

An Adobe Acrobat Reader edition with Semitic languages support was also used to extract the contents of PDF files and to convert them into HTML format. Characters that were unrecognized during the conversation, specifically in the Arabic text, were fixed as much as possible with their originals. For instance, the word النظام (meaning: system) had been altered at some positions to the unrecognized word ماظنك. Nevertheless, there was a major obstacle, which causes several Arabic PDF files to be dumped. This was the fact that many Arabic documents were found to be images placed in PDF files. This phenomenon results from the first phase, when the original editable versions of these Arabic documents were converted to PDF, using conversion tools. However, the phenomenon of image texts is common in Arabic files.

Unless a professional Optical Character Recognition (OCR) tool is used to extract text from these files, such image documents will be useless or handled as junk because it cannot be indexed in any textual IR system. However, most Arabic OCR software systems are commercial and are not available for free. Moreover, most of them, especially the few non-commercial ones, have some limitations, which in turn result in creating very noisy documents, for examples, many fonts are not recognized, unrecognized words, splitting letters of a single word, having problems with diacritized words and indentation of paragraphs.

In order to avoid this difficulty with respect to MULMIXEAC test collection, attempts were made to contact books' authors, especially publishers, in order to provide plain and editable text versions of the image documents. Unfortunately, responses were very low using this procedure. Hence, such documents in the MULMIXEAC were considered as junk that will not be parsed. Meanwhile, the OCR is beyond the scope of the work presented in this thesis. To sum up, it appears that much data in Arabic computer science is still not available in appropriate electronic formats and/or remains as hard copies. At Arabic universities, you may find a significant number of references/textbooks in Arabic (mixed Arabic and English) in hard copies but on the Web you cannot get a soft copy easily.

Throughout this stage of cleaning the collection, documents were also tagged, to some extent, with special tags for referencing purposes and for simplifying their collective representations. The name of the student who downloaded/wrote the document was used as well as and his academic level if the document is downloaded manually - otherwise the phrase 'automatically downloaded' was used.

Although the document collection is not annotated with metadata or part-of-speech tags, most of its categories within the computer science discipline were identified as much as possible from the early

phase. This is mainly done to help in creating a corpus with a query set having suitable and somewhat balanced coverage of the computer science discipline but in terms of the collected collection. The utilized classification for the process was the ACM CCS (Association for Computing Machinery Computing Classification System 1998)[16]. The identified categories in the collected documents using this classification include for examples, D.1 Programming Techniques, D.2 Software Engineering, D.4 Operating Systems, H.2 Database Management and C.2 Computer-Communication Network.

Duplicates were manually removed as much as possible. Beside human observation this duplicates removal was performed by issuing some random queries to the collection, after it was indexed as will be illustrated later, and tracking both automatically and manually gathered documents with approximate similar scores using only the (TF * IDF) standard scheme.

Cross-lingual Run-on words between Arabic and English were also recognized in mixed documents and fixed as much as possible. The run-on words (Buckwalter, 2004) problem was originally identified in monolingual Arabic documents – as illustrated in section 3.2.1. However, in this work, which is mainly multilingual, the problem occurs when the preceding word is immediately concatenated to the word that follows, while both words are different in their languages. For instance, the word 'الSemaphore' (meaning: the semaphore) is a cross-lingual run-on word because it is a concatenation between the Arabic definite ال (meaning: the) and the English word semaphore. However, in its valid form, the word should be written in two different words, the first is in Arabic and the second is in English - ال and semaphore, respectively. In multilingual documents run-on words in two different languages is a severe problem because it probably causes the IR system to stem the run-on words with the wrong stemmer, which is usually the Arabic stemmer, although the English word is the most significant among the constituent words in a certain run-on word.

In scientific Arabic documents there is always a higher possibility to write certain terminology in different regional variants. Regional variants in the Arabic texts in the collection were preserved as they appear in documents, although a significant proportion of the Arabic technical terms were found to be inconsistent and in different regional variants. Table 5.2 on the top of the next page shows a sample of these regional variations in the collection.

English Term	Arabic Term	English Term	Arabic Term
Linked List	القائمة المتصلة	Object Oriented Programming	البرمجة الشئية
	السلسلة المتصلة		البرمجة الكائنية
	اللائحة المترابطة		البرمجة موجهة الأهداف
Deadlock	الجمود	Normalization	التبسيط
	الإقفال		التسوية
	التقاطع		التطبيع
	الإستعصاء		

TABLE 5.2: Some regional variants in the collected document collection.

[16] http://www.acm.org/about/class/ccs98-html

The decision of preserving regional variants close to their original appearance was taken after a brief discussion with an expert in CLIR ("[Douglas Oard, personal communications, 2011]") with an eye to the fact that the variety of Arabic terminology is a realistic phenomenon on the Web that could not be avoided. Hence, any alteration (by unification) to this regional variation into a single term may lead to a biased conclusion about a certain algorithm when moving to realistic environments. Furthermore, such varied terms are very useful. For example, some researchers employed this phenomenon for creating a multilingual translation lexicon with regional variations in Chinese (Cheng, et al., 2004).

After these cleaning and processing procedures, a roughly cleaned and a raw document collection (Version A) of size 5.10 GB with its original contents (i.e. figures, formulae, tables) in clean HTML files was obtained.

5.2.1.2.2 Second Pass Cleaning

On the second pass of processing, which is devoted to creating the textual version (B) of the document collection, which is a pure text, in HTML format with HTML extension, an application program based on the distinguished HTML parser *Jericho* [17] was written so as to parse and filter pages in the different HTML formats. Jericho, which is an open source library, allows both analysis and high level manipulation of HTML files while at the same time it re-generates verbatim unrecognized or invalid HTML. Jericho also has the ability to recognize all types of server tags (ASP, JSP, PSP, PHP, etc) and, thus, HTML files can be parsed properly even when such server tags are included. Additionally, Jericho is also able to handle large files, approximately 2 MB or more, in term of streaming. This is important for parsing the collection. Thus, Jericho was used to remove HTML tags, figures, formulae and tables and the raw text was only preserved. During this stage, punctuation was also removed and unnecessary white spaces between words, which sometimes occur during conversion, was compacted.

The case-folding in English documents and English parts in multilingual documents were kept. Case-folding is important for the IR process, but in corpus-based analysis it may have a negative impact because it affects frequencies of words and the total number of distinct words and thus such normalization may bias some concluded results about a certain corpus. But, a very limited normalization process for Arabic documents and Arabic parts in multilingual documents was carried out. Specifically, kasheeda (see section 3.3.1) was normalized by removing the letters that were included purely for elongation (e.g., التجميـــــــــــع becomes التجميع). Diacritical marks (weak vowels) were also removed. However, diacritics in scientific Arabic documents are very rare but they were found in some typical documents.

Since the document collection is in two languages, encodings were different. For instance, Arabic pages were found in different encoding schemes. For examples, cp 1256, cp437, ISO8859-6, Windows 1256 and UTF-8. Therefore, all documents and pages were converted into a single common encoding (Unicode) that prepares them for indexing and analyzing by tools.

[17] http://jericho.htmlparser.net/docs/index.html

Following this, every word/phrase/portion/ paragraph - depending on how much a document is mixed - in documents was marked with a language tag attribute using a simple language identifier. This is essential for preparing the texts for multilingual indexing and it would help to identify the correct stemmer during the indexing phase. If a given document is in a monolingual language, the attribute "lang" is added to the body tag of the html file, e.g. <body lang ="en">; otherwise the "lang" attribute is added to a paragraph tag <p> in order to show that this portion of text is in a specific language, e.g. <p lang= "ar">. The former is used for monolingual documents while the latter is used for multilingual documents. Figure 5.1 shows a multilingual document after being processed, whereas Figure 5.2 illustrates another document that is being viewed by an Internet explorer. Arabic is written and read from right to left. Thus, insertion of English words sometimes makes sentences appear a little confused.

```
30 <p lang="en"> DES، </p>
31 <p lang="ar"> طُوِّر ثلاثة أساتذة جامعيون نظامَ تشفير آخر أطلقوا عليه اسم </p>
32 <p lang="en"> RSA، </p>
33 <p lang="ar"> ويستخدم هذا النظام زوجاً من المفاتيح مفتاح عام </p>
34 <p lang="en"> public key، </p>
35 <p lang="ar"> ومفتاح خاص </p>
36 <p lang="en"> private key </p>
37 <p lang="ar"> قد تم اختراقه فيما بعد وبقيت الحال على ذلك حتى قام فيل زيمرمان
38 <p lang="en"> Phil Zimmerman </p>
39 <p lang="ar"> عام بتطوير برنامج تشفير يعتمد نظام </p>
40 <p lang="en"> RSA، </p>
41 <p lang="ar"> لكنه يتميز باستخدام مفتاح بطول بت، ويُدعى برنامج الخصوصية المتفوِّقة
42 <p lang="en"> Pretty Good Privacy PGP </p>
43 <p lang="ar"> نُسخة مجانية، وهو من أكثر برامج التشفير انتشاراً في وقتنا الحالي
44 <p lang="en"> </p>
45 <p lang="ar"> ما هو التشفير </p>
46 <p lang="en"> encryption </p>
47 <p lang="ar"> ينها من عبث المتطفلين والمخربين واللصوص وتُستخدَم المفاتيح في تشفير
48 <p lang="en"> encryption </p>
49 <p lang="ar"> الرسالة وفك تشفيرها </p>
50 <p lang="en"> decryption </p>
51 <p lang="ar"> ة التشفير على عاملين أساسيين الخوارزمية، وطول المفتاح مقدّرا بالبت
52 <p lang="en"> bits </p>
53 <p lang="ar"> ير الرسالة وفك تشفيرها ويتفق الطرفان في البداية على عبارة المرور
54 <p lang="en"> passphrase </p>
55 <p lang="ar"> يستخدم المستقبل عبارةُ المرور نفسها من أجل فك شيفرة النص المُشفَّر
56 <p lang="en"> cipher text or encrypted text، </p>
57 <p lang="ar"> إذ تترجم البرمجيات مرة أخرى عبارةُ المرور لتشكيل المفتاح الثنائي <
58 <p lang="en"> binary key </p>
59 <p lang="ar"> ر إلى شكله الأصلي المفهوم ويعتمد مفهوم التشفير المتماثل على معيار
60 <p lang="en"> DES </p>
```

FIG. 5.1: A processed mixed document after being automatically generated in HTML format.

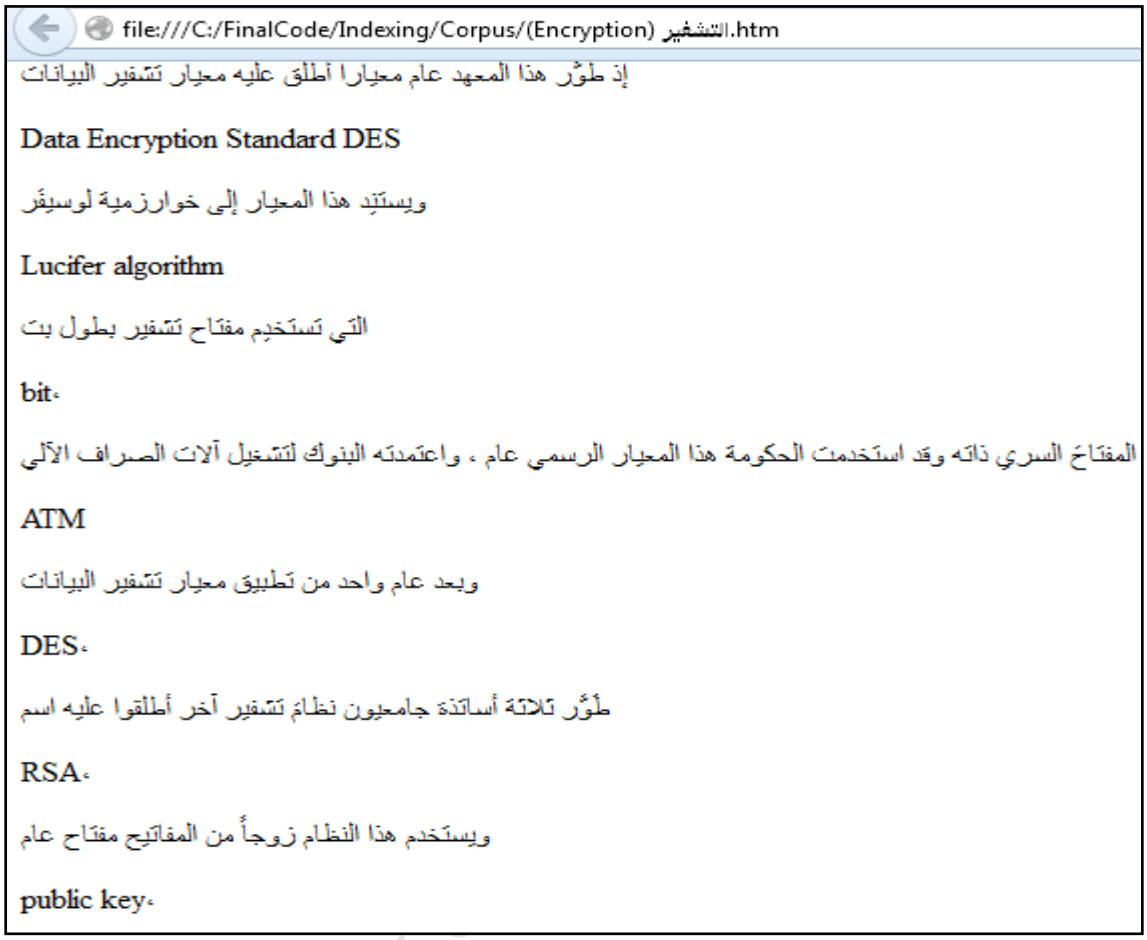


FIG. 5.2: A processed mixed document viewed in an Internet explorer.

At this point, these steps result in creating cleaned and purely textual documents, which are all placed in HTML format with a single codeset, with a size of 797 MB (0.8 GB). Thus, two versions of the document collection had been prepared: version (A) and version (B). Basically, the same texts are primarily included in both the two versions but they are different in their formats and layouts.

5.2.1.3 Collection Statistics

In order to obtain the essential information needed for the collection analysis, and also for experiments reported in this thesis, the *Lucene* IR system[18] was used. Lucene is an experimental information retrieval system that has being extensively used in previous editions of the CLEF, NTCIR and TREC joint evaluation experiments. The Apache Software Foundation[19]describes Lucene as a high-performance search engine with many full-featured libraries to process and manipulate texts. Furthermore, Lucence has the ability to index and retrieve files in the Unicode encodings. It is entirely coded in Java with many

[18] http://lucene.apache.org/core/index.html

[19] http://www.apache.com

powerful query types. The size of index in Lucene is roughly 20-30% compared to the size of text to be indexed. The Lucene has a diagnostic tool known as Luke[20] that is able to access indices that are being created by Lucene. Through Luke, it is possible to: browse documents; display frequent terms; analyze search results and optimize the index. Thus, using both Lucene and Luke, all documents in the document collection were indexed and analyzed. Since word boundaries in both Arabic and English are often set using white spaces, tokenization was performed on this letter as well as punctuation marks. In fact, text analyzers, which are mainly based on this word boundary, for both languages are provided in Lucence.

During the indexing process, appropriate terms are extracted. Firstly, four logical *fields* are defined. These fields are: <*TITLE-Arabic*>, <*CONTENTS-Arabic*>, <*TITLE-English*> and <*CONTENTS-English*>. The term field means a logical unit used by the Lucene IR system to populate document data and usually implementation of fields is a developer's decision, rather than a Lucene decision. In the experiments reported in this thesis, the <TITLE-Arabic> and <CONTENTS-Arabic> fields were used in the index for populating either texts in monolingual Arabic documents or the texts of Arabic portion(s) in mixed documents. The <TITLE-English> and <CONTENTS-English> fields were used for the monolingual English documents or the English portion(s) in mixed documents. Thus, depending on a document's language(s) some or all the fields may be utilized. However, in most IR experiments, and most CLIR, only two fields are used because of the underlying assumption of monolingualism in both documents and queries.

After creating the index of MULMIXEAC in the way illustrated, some statistics, e.g., number of monolingual documents and average number of words/document, about the document collection were extracted. Table 5.3 on the next page shows these statistics.

Description	Language(s) details	Numbers	Total
Number of documents	Monolingual English	51,217	69,184
	Monolingual Arabic	483	
	Multilingual(mixed Arabic and English)	17,484	
Number of words	English tokens	37,169,213	41,852,937
	Arabic tokens	4,683,724	
Number of distinct words	Distinct words in English	512,976	675,008
	Distinct words in Arabic	162,032	
Average number of words per document		605	

TABLE 5.3: Statistics for the MULMIXEAC collection. Figures are computed without stemming.

From the table it is observed that English is still the dominant language in common computer science, at least in terms of preferences of Arabic scholars. This is observed from the high number of English

[20] http://www.getopt.org/luke

documents, although the documents had been arbitrarily collected by native Arabic-speakers directly or indirectly by just providing a URL. However, the dominance of the English language on the Web is confirmed by several studies (i.e. Miniwatts Marketing Group, 2012). Monolingual Arabic documents on computer science are very scarce. In particular, the number of monolingual Arabic documents in the collection is only 483. This is due to the fact that Arabic speakers, especially scholars, do not know the proper translations or exact meanings for most terminology in their native language as well as the fact that they often use English terms instead of precise Arabic scientific terms.

It is also clear from the same table that mixed documents in the document collection are relatively few, compared with documents in English, at least in terms of discovery by the methods employed for collecting the data. In particular, the number of mixed documents is approximately one-third of the number of monolingual English documents. One reason behind this is the imaging characteristic, in pdf files, of scientific Arabic documents as discussed in the section 5.2.1.2. However, the proportion value of mixed documents to monolingual English documents, meaning the one-third, is much higher than what really appears on the Web (Miniwatts Marketing Group, 2012).

In the table also, it is noted that Arabic words (4,683,724) are relatively few when they were compared to the high number of the (37,169,213) English word in the collection. The total number of tokens in the document collection is relatively high (approximately 42 million words). This is true when it is compared to number of tokens in many standard collections, especially the Arabic ones. For instance, the 2001 LDC Arabic AFP collection contains 76 million tokens (Graff and Walker, 2001), which is not that much larger than words in the MULMIXEAC test collection, despite the big difference in number of documents in each collection. The number of documents in the 2001 LDC collection is 383,872 (about 5.5 times larger than the constructed document collection). Another example is the third edition of the Arabic Gigaword (Graff, 2007), which is also released from the LDC. The corpus contains 600 million tokens (about 14.3 times larger), but with two million documents (about 28.9 times larger than the number of documents in the MULMIXEAC document collection).This phenomenon of larger number of words in the created document collection is mainly caused by its genre type. Usually scientific documents are expected to be relatively longer than those in newswire because they may cover a specific topic intensively. In contrast, news genre often includes a constrained policy for length of articles and documents usually overuse regular words taken from general purpose vocabulary, unlike words in scientific genre, which has a high variety level from one topic to another. Furthermore, since the document collection had been collected from different sources such as references, websites, books, online helps, students essays and articles, forums, patents, etc, many writing styles and diverse vocabulary will probably appear due to the different writers, who are also different in their writing levels and their professions. Additionally, sample codes also influence the number of distinct words. Nevertheless, sample code also may decrease distinct words due to exhaustive use of many reserved words, e.g., public and private.

The same phenomenon of higher numbers is also observed when the unique words were extracted. In particular, the number of unique words in the corpus (675,008) is significantly high compared to those in many standard collections. For example, the number of distinct tokens in the 2001 LDC Arabic AFP corpus is only 666,076, considering the big difference in the number of documents. Alotaiby, et

al.,(2009) found only 2,207,637 distinct words, which is approximately 3.3 times the number of distinct words in the constructed document collection, in the third edition of the Arabic Gigaword. However, the number of documents here is two million (about 28.9 times larger). On a similar size of 600 million words taken from the third edition of the English Gigaword (Graff, et al., 2007), the study of Alotaiby found also about 1,257,112 unique words only (approximately 1.9 times larger than the distinct words in the compiled document collection).

The significant number of distinct words in the document collection results from several reasons. First of all, in the English documents, this is mainly due to the difference in the genre between the constructed document collection on one hand and the standard collections, which are mostly taken from general domain news stories. With respect to the Arabic *token types* (unique words), the high number of distinct words is caused by two major reasons, beside those which were provided above for the English language. Firstly, it was caused by the wide regional variety of the Arabic vocabulary in computer science across the Arabic-speaking. This fact has an important impact in raising the number of token types.

The second factor that causes an increase in the number of distinct primarily emerges from the language itself. Characteristics like Arabic grammatical rules, orthography, large number of affixes and the use of synonyms in writing style result in many unique words and consequently they affects the total number of distinct words in the document collection in general and total number of words in Arabic words in specific. Table 5.4 shows some examples for such varied Arabic words along their English counterparts, although their stems are similar. For instance, for the word حلقه (meaning: Loop) more than 15 different occurrences in the collection are counted due to the different attached affixes at the beginning, the middle and at the end of the word. Note that corpora are often analyzed in terms of words without stemming.

Arabic Term	Main English Counterpart	Arabic Term	Main English Counterpart
يرث	Inherit	المتغيرات	Variable
يرثهما		يتغير	
يرثه		متغير	
حلقه	Loop	الأشجار	Tree
بحلقات		الشجيره	
حلقة		الشجرة	
		الشجيرة	
مركزي	Centralized	تعويض	Compensation
مركزية		تعويضيا	
مركزى		التعويضات	

TABLE 5.4: Examples for some Arabic words in the collection. Each individual group has the same root stem.

Furthermore, the run-on words problem contributes also to the increase of Arabic distinct words, although the total number of words is relatively low.

Arabic Word	Frequency	Arabic Word	Frequency
مازالت	97	ما زالت	27
مازال	71	ما زال	21
ومازال	13	وما زال	10
ومازالت	8	وما زالت	12
فمازالت	6	فما زالت	7

TABLE 5.5: Examples of the different frequencies of the run-on words ما زال in the collection.

Table 5.5 shows the several forms of the imperfect copula verb (see section 3.2.1 in Arabic IR review when the run-on words problem was discussed) ما زال, which means is still, in the collection attached to different affixes, for examples to the letters: و and ف at the beginning. These different forms result from the monolingual run-on words problem between the word ما and the word زال and the morphological rules of Arabic. Additionally, spelling errors contribute to these several forms. For instance, spelling errors are very prevalent in the Arabic language, in general, and in the collection as well. For example, the mis-spelled words مازالت and مازال are three times more frequent in the collection than their correct counterparts ما زالت and ما زال, respectively. Such types of orthography as well as the typographical errors may invalidate the statistics of the collection, but yet this is what actually appears on the Web.

It is also obvious from the statistics in Table 5.3 that Arabic has more distinct words than English. Particularly, the number of distinct Arabic words in the 4,683,724 words is 162,032, whereas for the 37,169,213 English word it is only 512,976. Despite the significant difference between the number of words (the number of the English words is about 7.9 larger than the number of the Arabic words), the occurrence of distinct Arabic words represents approximately one third of the total number of the distinct English words. This is due to the same reasons, which have been listed above.

From the statistics in Table 5.3, it is also observed that the average number of words per document without stemming is relatively high (605 word/document). This is typically true when it is compared to standard collections such as the AP (Associated Press) newswire documents from 1988-1990 (TREC disks 1-3). The average number of words per document, which is computed without stemming, in this collection is 474 (Croft et al., 2010). This is relatively shorter when it is compared to the average number of words in Table 5.3 (the ratio between the two averages is 1.28), bearing in mind the differences in sizes. In the third edition of the Arabic Gigaword, the average number of words per document is nearly 300 whereas it is 419.6 for the same edition of the English language, but with 6 billion words and more than 7 million documents in monolingual English. Yet, both are lower on their average number of tokens per document in the MULMIXEAC collection. This is mostly related to reasons that were discussed above, particularly, those related to the nature of newswire collections, such as general purpose vocabulary dominance and policy of article length.

5.2.1.4 Collection Assessment

Usually, whenever a corpus / test collection is collected, its major features and its statistical nature should be explored. This is because features of a certain collection may have a major impact on the effectiveness of the developed IR techniques, which make use of the collection being created. Furthermore, general characteristics about a corpus will indicate whether the corpus is valid and appropriate to serve as a test-bed.

Accordingly, several measures and tests were proposed to evaluate the adequacy of texts in corpora (Kilgariff, 1997; Oliver and Berglund, 2002; Benajiba and Rosso, 2007; Baroni and Ueyama, 2006). This is especially true in linguistics and NLP fields. Such measurements include, but are not limited to, average word/sentence/paragraph length along with their distributions, richness of collocations, Kullback-Leibler distance measure for validating complexity, variety and correctness of word frequency distribution, checking homogeneity using Chi-square, detection of near duplicates using n-grams and many other natural language processing techniques.

One of the most important measures to study characteristics of texts in corpora is the statistical models of word occurrence. Such models are important and provide developers with deep sense or understanding of ranking algorithms and indexing techniques (Croft, et.al, 2010). For instance, significance of words in documents is considered with its frequency, whereas its importance is determined by word distribution statistics using the collection.

Although statistical measures and probability tests depend on the type of the corpus, its language, size and the task that the corpus is being used in, there are some important measures that should be applied to any type of corpora (Benajiba and Rosso, 2007). These are frequency distribution of words, vocabulary growth and token-to-type ratio. This is what the next section will discuss.

5.2.1.4.1 Zipf's Distribution

From statistical point of view, the distribution of frequencies of words in text is predicted to be very skewed (Croft, et. al, 2010; Manning and Schütze, 1999). This means that only small number of words, usually the most common, would have very high frequencies, whereas many words would have low frequencies. Thus, frequencies reduce rapidly with their ranks after the frequencies of the most common words. This statistical distribution is usually described by the *Zipf's law*, which is a commonly used model for describing the frequency distribution of words in a language or a collection. In particular, the law is used to predict the relationship between word frequencies and their ranks. Given a corpus/ document collection in a natural language, Zipf's law states that the frequency *freq* of any word in a collection (collection frequency of a given term) is proportional to the inverse of its position in the word list or its rank *rank* in the same collection. Alternatively, the frequency of a word *freq* times its rank *rank* is approximately a constant *k*:

$$k = freq * rank \tag{5.1}$$

Ideally, when *log*(*freq*) is drawn against *log*(*rank*) in a graphical representation, a straight line with a slope of -1 is obtained. However, since the law is a statistical model, it is expected that its prediction would not probably be exact, but it does this for most words in the collection on average. If the frequency *freq* of a given term *i* at rank $rank_i$ is substituted by its probability *Pr*(*i*), which is computed as the frequency of that term over the total number of terms in the collection, then equation 4.1 can be written as:

$$k = Pr(i) * rank_i \quad (5.2)$$

Where $rank_i$ is the rank of term *i* and *k* is a constant for the collection.

To apply Zipf's law in the MULMIXEAC collection, the *unigram language model* (the frequency of each token) is employed. Hence, unique words with their ranks and frequencies are firstly extracted.

* Freq. = Frequency

	Arabic				English			
Rank	Token	Freq.	%	Pr(i)* $rank_i$	Token	Freq.	%	Pr(i)* $rank_i$
1	من	119147	0.285	0.003	the	2242811	5.366	0.054
2	في	111064	0.266	0.005	of	900770	2.155	0.043
3	على	72013	0.172	0.005	to	876674	2.097	0.063
4	و	65141	0.156	0.006	a	859291	2.056	0.082
5	أن	40445	0.097	0.005	and	694178	1.661	0.083
6	أو	36467	0.087	0.006	is	603148	1.443	0.087
7	إلى	34116	0.082	0.006	in	592150	1.417	0.099
8	التي	30574	0.073	0.007	for	409930	0.981	0.078
9	هذه	30505	0.073	0.007	The	348493	0.834	0.075
10	هذا	26769	0.064	0.007	this	278881	0.667	0.067
11	عن	25897	0.062	0.007	be	252061	0.603	0.066
12	البيانات	25549	0.061	0.007	are	224024	0.536	0.064
13	مع	19788	0.047	0.008	as	217454	0.52	0.068
14	هو	18927	0.045	0.007	you	216721	0.519	0.073
15	لا	18610	0.045	0.007	by	213440	0.511	0.077
16	ما	17956	0.043	0.007	it	210846	0.504	0.081
17	كل	15463	0.037	0.007	or	204456	0.489	0.083
18	الذي	15459	0.037	0.007	with	204368	0.489	0.088
19	هي	14500	0.035	0.007	an	200234	0.479	0.091
20	ثم	14217	0.034	0.007	on	188314	0.451	0.09

TABLE 5.6: The most frequent 20 unigrams in each language (top 40 words) in the gathered collection.

Table 5.6 illustrates the most frequent 20 words in each language in MULMIXEAC along with their frequencies and their percentages of appearance (converted probability). Apparently, in both the two languages, most frequent words are prepositions, particles, definite articles and stopwords, however they

have a big task in attaching words together. Nevertheless, in the Arabic list of most common words, there is only one word that does not belong to such a set of particles, that is the word البيانات (meaning: data).

It is also noticed that the frequencies for the most top 20 words in Arabic are lower than their peers in English. This is due to the fact that the majority of documents along with words in the document collection are in English. It is also clear that the frequencies of words begin with very high values (see the frequencies of the words at the 1st rank in each language). Then, the frequencies in both languages begin to decrease rapidly as new words are ranked and after the appearance of the few frequent words (see the frequencies of the words at ranks 19 and 20, for examples, in each language and compare them with those in the 1st rank).

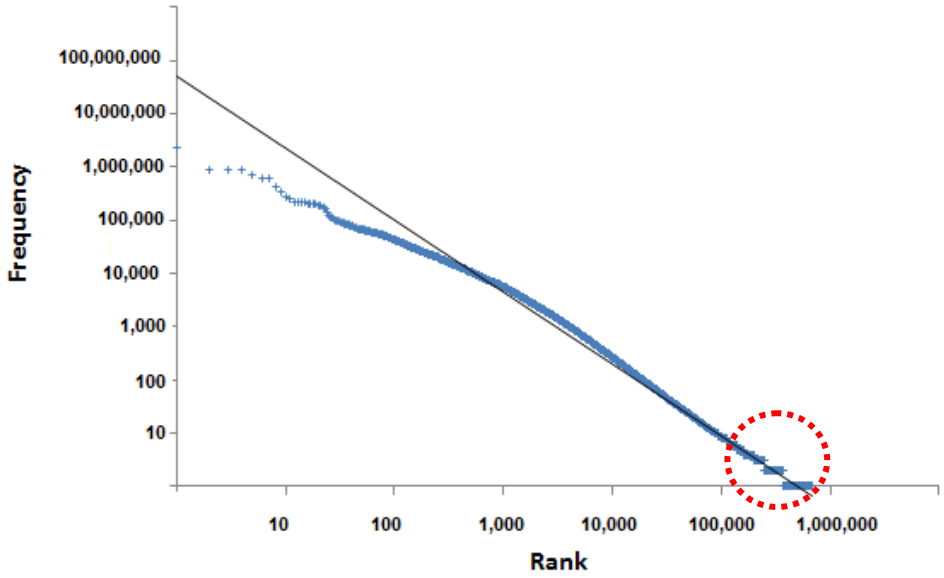


FIG. 5.3: A log-log Zipf's curves (actual and predicted) for the 675,008 unigrams in the collection.

Figure 5.3 shows Zipf's law applied to the unigrams in the MULMIXEAC document collection. The blue curve illustrates the actual relationship between the word ranks and their frequencies, whereas the straight line in black shows the predicted relationship between them using the least square method that calculates the best fitting line to data.

Form the figure, it is observed that the actual Zipf's curve and the fit of the data to the law from the collection is quite close. In particular, the curve clearly reveals that frequencies of words decrease rapidly with rank, meaning that distribution of words in the corpus is skewed. Thus, it appropriately predicts words frequencies in the collection with a slope of -1, approximately, except for some most frequent words. In particular, the figure reveals that the actual curve is inaccurate for, approximately, the words in the first 300 ranks.

From the dashed circle in Figure 5.3, it is noted that the majority of the words in the document collection are *hapaxes* (words that occurring only once). This is due to the fact that MULMIXEAC is a special collection, whose vocabulary is expected to be diverse. Additionally, the collection is in computer science, meaning that many functions, methods and reserved words are provided, e.g., Math.sqrt(). The Arabic

morphology contributes to these hapaxes, as well. However, hapaxes in corpora are often considered as important words because they will acquire higher weights whenever they appear in a query.

5.2.1.4.2 Vocabulary Size Estimation

The size of the vocabulary in corpora is usually estimated by *Heap's law*. The Heap's law is used to predict vocabulary growth in a certain collection (Croft, et. al, 2010; Manning, et al., 2008). In particular, the law is used to predict the vocabulary size (number of distinct words) as a power law function of a collection size (number of words). This power function states that the number of distinct words d in a given collection with M words is approximately *sqrt(M)*. More formally, the relationship between the size of the corpus, which is denoted by M, and the size of the vocabulary, denoted here by d, in the corpus is:

$$d = a * M^{\beta} \tag{5.3}$$

where a and β are parameters that vary for a certain corpus to another. For this formula, Heap's law predicts that new words would result in a rapid increase in vocabulary when the collection size is small. However, when the corpus size increases, more new words would still increase the vocabulary size but, at slower rates (Croft, et al., 2010). The typical values for the parameters a and β are: $10 \leq a \leq 100$ and $\beta \approx 0.5$ (between 0.4 – 0.6). The reason behind the quite large range in the variability of the a parameter is that it depends on factors like stemming, case-folding and spelling errors (Manning et. al, 2008). For instance, spelling errors are directly proportional to the growth rate.

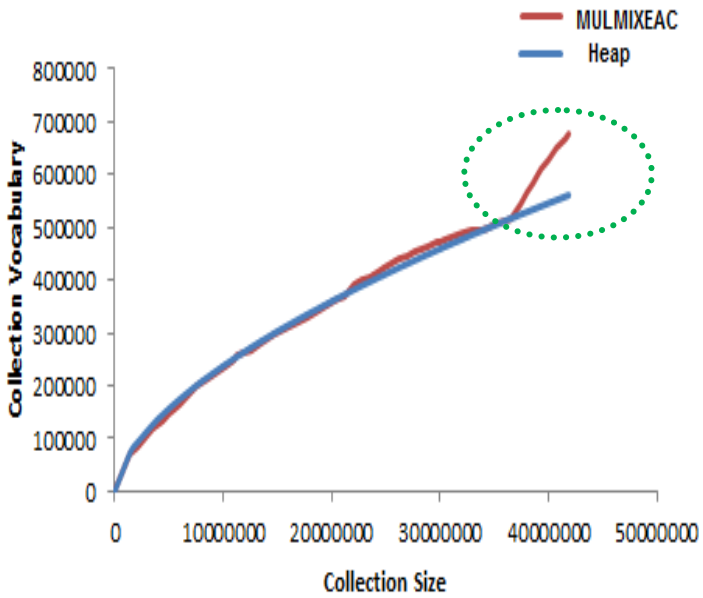


FIG. 5.4: The predicted and the actual vocabulary growth in the corpus using the Heap's law.

Figure 5.4 shows both the predicted vocabulary growth, corresponding to the blue curve in the figure, and the actual Heap's curve (with $k = 15$ and $\beta = 0.6$), represented in the figure with a red curve, for the

MULMIXEAC corpus. The figure shows clearly that the growth of the vocabulary in the corpus is a good fit. Nevertheless, as the number of words reaches approximately more than 35 million (which is approximately the number of the English words) it begins again to increase rapidly, instead of steadily - see the green dotted oval in Figure 5.4. There is a possible explanation for this observation, which is mainly caused by the multilingual characteristic of the corpus.

It is usually observed that English documents are named with an English file name, whereas the names of Arabic documents are mostly in Arabic, although there are many names that are mixed (begin with Arabic or English letters). This fact causes English documents, and thus English words, to be ranked ahead and before the Arabic documents, as the Arabic letters often have a higher codeset and thus, lower ranks, when the application program begins to accumulate both the number of words and the distinct words. Thus, when English vocabulary begins to grow at slower rate (after the rapid increase at the beginning), Arabic documents appear and they begin to accumulate their vocabulary and thus, the curve begins to jump, approximately after more than 35 million words. Meanwhile, it is possible to randomize the document selection after applying a numbering mechanism for documents. However, another scenario was applied, that is to implement Heap's law for each language separately in the MULMIXEAC.

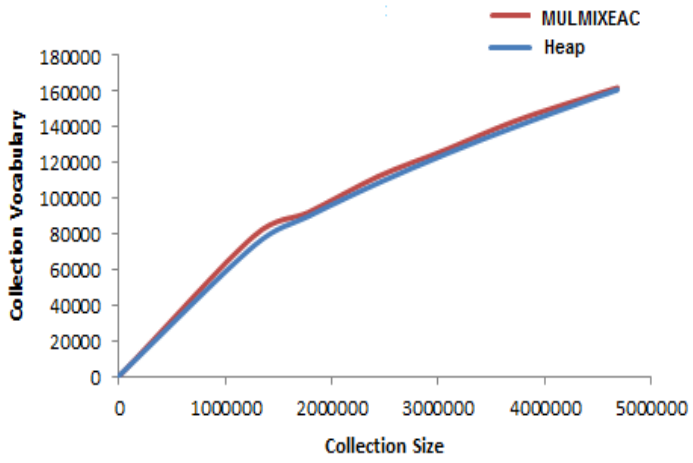


FIG. 5.5: The predicted and the actual vocabulary growth for the Arabic texts in the corpus using Heap's law.

Figure 5.5 and Figure 5.6 show the predicted vocabulary growth along with the actual growth in the corpus for both Arabic and English texts, respectively. The curve for the Arabic language is a good fit. This good prediction is clear at different points. For instance, in the first 1,368,222 words in the corpus, Heap's law estimates that the number of distinct words is 76,880, whereas the actual value is 77,991. Furthermore, in 4,683,724 words, Heap's law predicts 160,870, whereas the actual number is 162,032, which is very close to the predicted value.

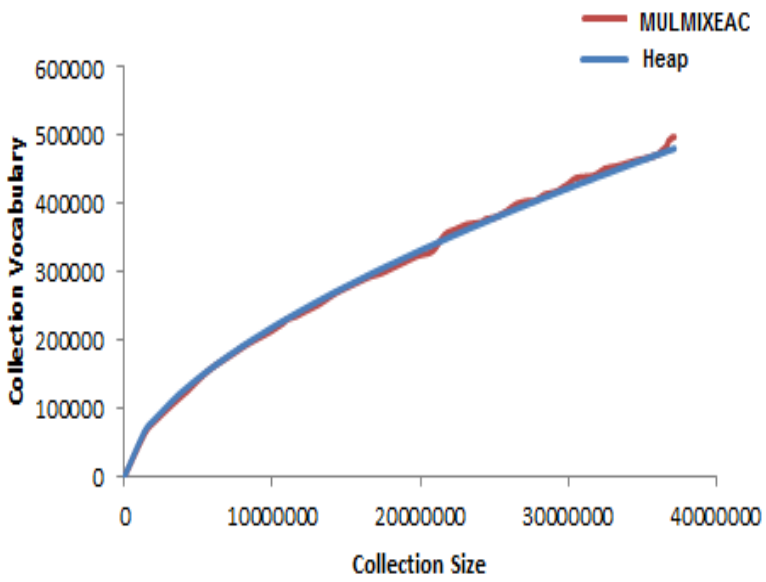


FIG. 5.6: The predicted and the actual vocabulary growth for the English texts in the MULMIXEAC corpus using Heap's law.

For the English text the actual Heap's curve is also quite accurate. Nevertheless, the curve continues to be accurate and close to the predicted curve, even after approaching larger values (i.e. 30 million words), although the vocabulary growth is expected to drop at a certain point (i.e. when no new tokens are added to the accumulated vocabulary). This observation results from the fact that vocabulary in scientific documents in a certain track may be very varied from other tracks e.g., information retrieval field vs. human computer interaction field. Thus, the actual curve does not break down gradually. Instead it gets closer to the predicted vocabulary growth. However, this observation depends also on the type of the corpus as well as its size. All these findings confirmed what was concluded above when the entire corpus is analyzed together.

5.2.2 Query Set

Queries for experimental purposes can be created using different approaches. One realistic approach when moving from experimental systems to realistic systems is to use queries that are collectively represent queries posted by users of the target application (Croft, et al., 2010). Such queries may be acquired either from a query log from a similar application or from potential users directly. Such an approach (asking potential users for sample queries) provides more realistic results and fills the gap between the real environment and the environment of the developed algorithms. Moreover, the approach has being used in creating query sets in many well-known forums, for example TREC Query Track. Therefore, this approach is followed to create the query set for the MULMIXEAC corpus, although, the scenario here is somewhat different. This is because in the task of TREC Query Track, users are usually asked to submit examples for queries after being shown the texts of topics, which are the information

needs. For instance, in his study to examine variability of queries in the TREC-9 Query Track, Buckley(2000) attempted different approaches for constructing a query track test collection. The approaches asked users to write a query after showing them both the topic text and/or relevant documents. However, in creating a query set for MULMIXEAC another approach was applied, as will be discussed next.

5.2.2.1 Producing Initial Query Set

In the case of multilingual and mixed queries and documents, such an approach of showing both topics and/or relevant documents would probably directs the potential users, who are usually asked to create the queries, to write their queries using the same language of documents (i.e. monolingual Arabic documents/topics would probably result in producing monolingual Arabic queries). This is because the users would likely be influenced by the texts shown to them.

To avoid such behaviour, the process of generating queries to MULMIXEAC was semi-blind. Firstly, a group of 50 students at different academic levels at computer science departments in two Arabic universities were selected as potential users. The sample group also include some tutors and Teaching Assistants. Participants are native Arabic speakers with medium level in English. Each potential user in the group was requested to provide a set of 10 queries on common computer science vocabulary. It was intended that the number of potential users should be large, whereas the number of queries should be relatively small. This would likely result in better diversity (instead of 20 queries per user for 25 participants).

In order to implement the blindness, the choice of the query language was deliberately avoided and, hence, participants could show their natural searching behaviours. Furthermore, there was no constraint about the query lengths. Before submitting the queries, participants were only shown the categories of the MULMIXEAC corpus, but without any pre-knowledge about the corpus itself. Again to avoid directing users to a specific language of queries, categories were presented in two parallel texts (Arabic and English). This will help to have good/adequate coverage for categories (and thus for topics) presented in the corpus. Additionally, using such an approach usually results in that queries will not return empty results when they are issued to the MULMIXEAC index. To this point, a raw set, which was semi-blindly produced, consisting of 500 queries was obtained.

5.2.2.2 Analyzing Initial Query Set

The sample scholars who participated in creating the queries used various search languages (in terms of Arabic or English or bilingual Arabic and English) for their information needs. However, the choice of the language(s) of queries submitted by a single participant is sometimes inconsistent. Some users prefer to use their native language, which is Arabic, whereas others use monolingual English and most of them posted queries in mixed forms (multilingual querying). In particular, more than 48% (240 queries) of the created set were expressed in multilingual forms. Monolingual English queries represented also a higher

proportion (about 42%), whereas Arabic the query proportion was only 10% of the total number of the submitted queries in the set. However, using the English language for searching in a scientific domain is very common, but it was not surprising that very few Arabic queries were submitted. A list of 47 queries is provided in appendix B as examples for the collected mixed queries.

A discussion was held with the participants about why they chose a specific monolingual and/or mixed language to approximate their information needs. For those who submitted mixed queries, most of the participants were limited by the modern vocabulary of computer science in Arabic or in the best case they would not want to miss valuable relevant documents due to regional variations. In appendix B, examples that include English words and phrases: Secure Socket Layer, Synchronized Methods, three tier architecture, stable marriage problem, inner-join and outer-join and divide and conquer algorithm. To confirm this fact, 25 of those participants were asked about the accurate translations for the English terms in their multilingual queries. As estimated, more than 72% (on average) of them could not provide popular/appropriate /understandable translations. For instance, although scholars understand well what the 'mutual exclusion' phrase is, they did not know its accurate translation. The participants who submitted mixed queries also discussed why they did not submit their mixed queries in the alternative form of monolingual English. They simply stated that such monolingual English queries will definitely result in retrieving English documents only (at least on the top 100 documents). Thus, they used mixed queries so as to retrieve both Arabic documents, which are usually mixed, and English documents and did not mind if the latter documents are retrieved at the lowest ranks because this is the only way that the current search engines provide for them and thus they are usually forced to traverse through the result lists. There is an important observation that the participants revealed - that is the majority of them stated clearly that in their searching behaviors using current search engines, they adapted themselves to begin usually their exploring of a certain retrieved list by starting with the first 1-10 documents (on average) then they immediately jump to the pages that contain monolingual English documents only. This fact is evidence of the necessity for devising algorithms that meet the needs of non-English users. Furthermore, the fact also reveals that much of the scientific Arabic content on the Web, at least in terms of computer science, is not of good quality, although such content is found but owned by publishers/institutions.

There is another interesting feature in the submitted multilingual queries. A few queries (less than 10) consisted of both the Arabic technical term and its corresponding English translation, e.g., فهرس التجميع clustering index. The Arabic phrase فهرس التجميع here is the translation of the English phrase clustering index. Users who submitted such queries might desire to obtain higher recall. Another example for such a query in appendix B is ' التطبيع normalization في قواعد البيانات ' (meaning: normalization in databases). In this query the word التطبيع is accompanied with its English translation normalization.

From the submitted queries, it was also observed that English portions of multilingual queries are often scientific/specialized. This is obvious in appendix B. They are significant for searching and strong candidates for hitting the most relevant documents. Furthermore, it is also noted that if an English part in a certain mixed query consists of two or more words, then it probably corresponds to a terminological phrase, for examples, data mining and entity and relationship model.

Arabic parts in multilingual queries, as shown in appendix B, were found to be general-purpose vocabulary, such as the word مفهوم, which means concept, with or without some stopwords, like the words: ما هو (meaning: what is), بين (meaning: between) and ال (meaning: the). Additionally, it was observed that the same Arabic general-purpose vocabulary/stopwords are usually overused. This characteristic is not present only in mixed queries, but also in monolingual Arabic queries.

With respect to the monolingual Arabic queries that had been posted, it was found that scientific Arabic terms in them are mainly unambiguous and meaningful. Usually such Arabic scientific terms are either a language name (proper noun), such as Java (meaning: java) or a term taken from English general-purpose vocabulary with a very straightforward and proper translation in Arabic (such terms cannot be translated to Arabic except in the provided translation), such as ذكاء اصطناعي (meaning: artificial intelligence) and هندسة برمجيات (meaning: software engineering). Nevertheless, usually such scientific Arabic terms are relatively few when they are compared with the large number of English terms, which have no proper translation or users do not know their precise translations. However, using only Arabic queries in scientific domains usually results in retrieving a relatively limited list of documents, due to a large number of synonyms, which could be easily missed, and orthographic variations, especially regional ones. Furthermore, many of the irrelevant documents are usually retrieved. This is due to the language morphology and the wide spread of the homographic words. For those who submitted English queries, the discussion with the participants leads to the fact that the decision was mainly related to their ability to express queries in the English language.

5.2.2.3 Producing Final Query Set

All the submitted queries were pooled into one set. Extra spaces, punctuations and capitalized letters in English words were removed. Duplicates and semi-similar queries, even those in different languages, were removed, but if there is a duplication between a monolingual query and a mixed query, the latter is chosen and is placed in the final query set. After these removals, there were 128 unique queries left. This high reduction in the number of queries was caused by the fact that students in similar classes usually share the same interests and ideas. Furthermore, since this research is on mixed queries, the majority of monolingual queries (about 260 queries) were also removed.

In section 2.4.2, it was illustrated using a reasonable number of queries, particularly 50, could tell whether an improvement of one algorithm is better than another and the difference in performance of 0.05, using MAP for example, will result in an error rate below 4%. Therefore, a cleaned set of 47 multilingual queries was chosen to represent source queries for the experiments on the MULMIXEAC corpus. The set represents approximately 36.7% of the final query set. The choice of the 47 queries was primarily governed by the categories in MULMIXEAC but with a primary goal of the developed applications, that is queries should be mixed. Before selecting the queries in the set, all categories in the MULMIXEAC collection were first assessed by staff members in computer science, who finally determines which queries are to be included in the set according to categories. Queries were numbered (DLIB001-DLIB047) for referencing purposes. Some sample queries are listed with their translations/meanings in

English in Table 5.7, whereas appendix B shows the whole query set. It should be noted that in spite of the possibility of using more queries, for example, by utilizing the complete set of the mixed queries or requesting potential users to provide more mixed query examples but, the relevance judgment is a significant constraint as it needs much efforts. The goal here is to use sufficient number of queries to make a significant difference between developed algorithms.

Query #	Query	Counterpart in English
DLIB01	Deadlock مفهوم الـ	Concept of deadlock
DLIB02	Secure Socket Layer ماذا نعني بالـ	What is meant by Secure Socket Layer
DLIB03	Interpreter و ال الفرق بين الـ Assembler	Difference between interpreter and assembler
DLIB04	Polymorphism شرح في الجافا	Explain polymorphism in Java
DLIB05	Entity Relationship Model مثال في	Entity and Relationship Model, Example
DLIB06	Data Mining تقنيات	Data Mining techniques
DLIB07	Synchronized Methods تمارين جافا	Tutorials on synchronized methods in Java

TABLE 5.7: Examples of some mixed queries (DLIB01-DLIB07) in the created query set.

Arabic Technical terms in the final list were very few. These Arabic technical terms, however, appeared mostly on long queries in which other English technical terms were also involved. For example, in the query numbered by DLIB04, 'شرح polymorphism في الجافا' (meaning: explain polymorphism in Java), the Arabic word الجافا (meaning: Java) is a technical term but, it also appears with the English technical term polymorphism.

Table 5.8 on the next page shows some statistics about the query set. The average number of words per query in the set, without stemming and stopword removal, was found to be 4.4 with 2.3 and 2.0 as the average number of words for English and Arabic, respectively. These lengths in multilingual queries are relatively longer than their peers in monolingual queries ("[Douglas Oard and James Mayfield, personal communications, 2011]").

Description	No.
Average no. of words per query	4.4
Average no. of words in Arabic per query	2.0
Average no. of words in English per query	2.3
No. of queries	47

TABLE 5.8: Statistics about the query set of the MULMIXEAC corpus, rounded to one decimal.

However, this is mainly caused by the multilingual characteristic of queries, whose usage would probably result in extra words. In particular, whenever users attempt to express their information needs (queries), they would likely be faced with a gap, which is incurred from the use of the two languages together, for example, mixed words between languages may not be consolidated enough to approximate users' needs.

Therefore, to fill this gap one might usually need extra words. Furthermore, the genre of MULMIXEAC has an impact also on the relatively long queries. In a scientific domain, as in computer science, users may submit relatively accurate queries in order to obtain good results. For instance, one might need to search about overriding of methods in Java language. Such users would probably submit a query like 'method overriding Java', instead of 'method overriding' only as many languages use the same concept.

5.2.2.4 Converting Query Set into Topic Files

The 47 queries were put into similar formats to the TREC *topics* (queries) in order to provide guidelines later for the assessors of the relevance judgments of documents with respect to queries in experiments. In most standard forums topics in the ad-hoc track are primarily presented in three structured fields' tags: title, description and narrative, as discussed in review chapter. In the work reported in this thesis, the same three fields were used, but three extra fields were added also. These are *originalQuery, domain* and *creator*. The originalQuery field stores the original mixed-language query in both Arabic and English. The creator filed is added for referencing purposes, i.e. the name of the creator of a certain query. For lawful purposes, a decision was taken to add the creator name only if his permission was granted ("[Douglas Oard, personal communications, 2011]"). The domain field is added for any further expansion of the corpus. In particular, the corpus is intended to be expanded in the future by adding many scientific Arabic documents from different genres of science. For such purpose, the domain field will be considered as the first level of categorization of documents. Currently, the domain field contains only the phrase 'Computer Science'.

All fields, except the originalQuery are presented in the topics file in parallel text language(s). Thus, there is a topic file in Arabic with a translated version provided for each field in English, but in another separated file. Therefore, there is an attribute in the *top* tag, namely '*lang*', used to specify the language of the topic file. Figure 5.7 illustrates a part of the English topic file of query number (DLIB001).

<top lang="en">
<num> Number: DLIB01
<OriginalQuery> مفهوم الـ Deadlock
<domain> Computer Science
<Title> deadlock concept
< description >
What is and how does the deadlock occur in operating system and what are the solutions provided to solve it?
< narrative >
Relevant documents must include details of what the deadlock is in operating

FIG. 5.7: Part of the English topic file of query number (DLIB001).

The originalQuery filed is the only field that is written multilingually in the two files and cannot be attributed to any language (i.e. Arabic or English). Translations of the topics files were prepared by some

five volunteers of a staff at computer science departments in two Arabic universities. Those members are experienced with a strong academic base in computer proficiency, since they are lecturers. They are also native Arabic speakers and they are fluent in English. A house-collected computer-based dictionary, gathered from different sources, was also used in order to unify translation of terminologies across topic files and to increase consistency. This is important for the Arabic translation of topics. The same dictionary was also used in experiments as will be illustrated later in the next chapter.

The contents/sentences of the six fields themselves were created and audited in a collaborative process between the original creators of queries and the same five staff members, who translated the topic files. This is especially true for the narrative field as it will be used for the relevance judgments.

5.2.3 Relevance Judgments

In order to determine how the relevance judgments were performed, it is important firstly to specify a retrieval task in terms of the application's goals. In particular, the kind of the required judgments along with the levels of relevance are essentially influenced by the type of the required retrieval task. For instance, for a certain task, multi-level relevance judgments may be more appropriate than binary relevance. However, binary relevance is the most dominant on collections used by the different editions of experiments (i.e. TREC), whereas the search task is the high recall, in which it is important not to miss information (Croft, et al., 2010). With respect to mixed-language queries and documents, result lists are often subjected to two major drawbacks that have their impacts on the required retrieval task. Firstly, there may exist a highly relevant monolingual document that is ranked at the lower part of a result list (because its score is computed from only a portion of the mixed query). Secondly, a poor relevance mixed document may be ranked higher, as its score is computed from the entire mixed query. From that perspective, it is important to distinguish between rankings of different documents so as to find out to what level of effectiveness the IR system does at retrieving the most relevant documents at top ranks (i.e. whether the highly relevant documents are ranked at the top of the retrieved list or not). Bearing in mind that users often tend to explore only the top documents, this probably makes a difference because a retrieval algorithm that ranks a certain highly relevant document at rank 2, is expected to be better than another retrieval algorithm which ranks the same document at rank 9. In such situations, tasks like high recall or precision at a predefined position (i.e. at p, where $p = 10$, for example) may not be adequate. This is because such tasks do not consider the rankings of documents. For instance, assume that there is only one relevant document in the top 10 returned documents. Using the precision at p, e.g., 10, the rank of this document in any position from 1 to 10 would be similar (Croft, et al., 2010). Thus, to overcome such a limitation and considering documents rankings, a more precise measure, and thus task, is needed. Apparently, the most appropriate retrieval search task for the work presented here is to emphasize highly relevant documents and whether they are ranked at higher positions. For such a task, multiple levels of relevance (graded relevance), which was discussed in section 2.4.2 in CLIR review chapter, should be employed. As discussed by many authors, e.g. Croft, et al., 2010, using garded relevance needs only few numbers of documents, typically 10, to be judged.

For the experiments reported here, a long discussion was held with 5 PhD holders, who are experts in computer science, native Arabic speakers and are fluent in the English language, to determine the number of degrees/points in the graded relevance that will be used to assess documents. Several documents in different themes were exhaustively examined. TREC standards as well as some valuable experiments (Kekäläinen, 2005) were also consulted. The creators of queries were consulted as well by presenting to them several documents with different relevance levels for their created queries. Bearing in mind the nature of scientific documents in computer science, it was concluded that the assessment should be done on a six-point scale (0-5). In a descending order, these points are as follow:

[5] *Highly relevant document*: The document discusses the subject and the arguments of the topic of a query comprehensively. If there are many aspects in the topic, then the document should cover all or most of them. Informally, the assessors estimated that a highly relevant document could be taught as a lecture of 45 minutes at least.

[4] *Suitable relevant document*: The document discusses the subject and the arguments of the topic of a query in a suitable way. Presentation is exhaustive in some parts whereas it is not in other parts. In the case of a multi-component topic, more than the half of the sub-components are covered. This is similar to saying that the document is definitely relevant.

[3] *Partially relevant document*: The document discusses the themes of the topic of a query partially. The majority of the presentation is not exhaustive and is covered briefly, whereas few parts (less than half) are covered in detail.

[2] *Marginally/very low relevant document*: The document only discusses the topic very briefly and in many cases it only points to some themes of the topic.

[1] *Possibly not- relevant document*: The document points to some themes in a very limited boundary that could be insignificant to the topic of a query. Nevertheless, it is not possible to say that the document is totally/definitely irrelevant.

[0] *Irrelevant document*: The document does not contain any information about the topic of a query.

Based on this six-point scale, documents were assigned relevance level values. However, since the multi-level relevance was used, the average discounted cumulative gain at top 10 documents was employed as a performance measure. The DCG was illustrated in section 2.4.3. Indeed assessing more documents is better than only 10 documents. However, besides the reasons provided in the prvious paragraphs, this is mainly because of that a considerable manual effort is required for such a task. Furthermore, the use of DCG measure usually emphasizes top ranked documents and thus, relevance only required to a certain rank, which is typically 10 (Croft, et al., 2010).

Relevance assessment files themselves, which contain relevance values for each query, were obtained by using a pool-like mechanism. In that mechanism, a given query in the query set was used to produce several ranked lists, which were obtained by the different algorithms reported in the evaluation chapter. For each ranked list, the 10 top documents were extracted and all documents were joint together into a single pool. Thus, the pool contains both relevant (even highly or marginally relevant) and irrelevant documents. Duplicated documents were removed and the list was shown in a random order to the assessor team, whose members judged documents one by one, with respect to the source query. The

team of the assessors, who were computer science scholars, contained two PhD candidates and four Master's students. The team was working under the supervision of a volunteer PhD holder. The team was carefully selected with essential criterion, beside the strong academic background, that each member should at least be assisting in lecturing/teaching tutorials/ doing (his/her) research in fields related to queries' topics. All of the team members were fluent in both Arabic, as a native language, and English. Nevertheless, the creators of the queries participated also in the relevance judgments of documents, according to their posted queries and as much as possible. As in TREC assessement for ad-hoc retrieval track (Kekäläinen, 2005), the assumption behind relevance is the topicality. Thus, each mermber of the assessors was asked to assess the pooled documents. Next, relevance garde of documents of all assessors were compared to each others. If the relevance grade of a particular query is varied, the ones who assessed that document are consulted again and a new volunteer assessor was requested to re-assess the document. Thus, the relevance judgments were exhaustively performed.

Table 5.9 illustrated the number of judged documents for all experiments reported in the evaluation chapter. As results of different algorithms were varied the judgments were relatively high. However, this is also due to that experiments were firstly conducted using vector space model before a decision was made to conduct them using okapi BM25 ("[Douglas Oard, personal communications, 2011]").

Description	No.
No. of judgments for all documents	1983
Average no. of relevance judgments per query	42.1
No. of queries	47

TABLE 5.9: Statistics about number of relevance judgments in the corpus, using different algorithms.

5.3 Summary

Most existing test collections, and most CLIR collections, are either concentrated on rapid use of general-domain news stories – written in a monolingual language or containing multiple monolingual corpora. Furthermore, specialized corpora lack many languages, including Arabic. In fact, such specialized corpora were released in few languages. Additionally, their query sets are essentially monolingual and/or mixed documents in them are processed as if they are in a single language. Therefore, in this chapter, a multilingual and mixed Arabic-English test collection, named MULMIXEAC, on common computer science vocabulary had being created. The collection is also synchronic. The corpus, which will be used as a benchmark for experiments in this thesis, is primarily gathered from the Web as using such a resource is cheap and allows building a large amount of data in any genre within a relatively short time. However, issues like dissemination of information and copyright permission, many noisy documents and un-editable and imaged documents, especially in the Arabic language, are major difficulties that prevent

collecting much larger data. Furthermore, the significant amount of both time and effort that was needed to build a corpus, in general, along with the planned target of completion of the MULMIXEAC collection within the assigned-one year duration are also major reasons, but yet the corpus is sufficient to run the experiments.

Text in the corpus was filtered, cleaned, indexed and examined in terms of statistics with an eye on comparing them to reference collections. Furthermore, some useful statistical tests were applied to the corpus. Evidence showed that the distribution of frequencies of words in the MULMIXEAC corpus is very skewed. Furthermore, it has shown that the vocabulary growth in the corpus is a good fit. This is important because if a one implemented only the predicted vocabulary growth on the corpus, then it can be estimated, roughly, that extra added documents to the corpus, in the future, will not significantly change its nature. In addition, it was observed that the Arabic language in the corpus has more distinct words than English, thus resulting in lower Token-To-Type (TTR) ratios.

The next chapter reports the experiments along with their evaluations.

CHAPTER

6 Evaluation

This thesis attempts to explore how to allow users to issue queries in a multiple languages (mixed) form to search across mixed and multilingual documents. At the same time the results must be ranked according to their relevance, rather than their exact match to these mixed queries, and regardless of the dominant language in the query words or documents and regardless of the user's ability to express concepts in a particular language. This chapter shows how such a language-aware/mixed-language IR system was evaluated. In particular, the chapter will evaluate the newly developed approaches, which were shown in the design chapter, for such a mixed-language IR system. This evaluation aims to show the substantial impact of using these proposed approaches on retrieval effectiveness and determine if they provide a significant improvement over some well-established baselines, including a monolingual run. To achieve this, two different sets of experiments were carried out using the MULMIXEAC test collection and the previously produced query set in chapter 5. The first set of experiments was developed in a centralized-based architecture. It is shown that using the latter approach may incur major drawbacks, especially when it comes to mixed-language queries and documents. Thus, if this mixed-language feature is not well controlled with regards to weighting of terms in the centralized architecture, the retrieval effectiveness may be hurt significantly. Accordingly, the first set of experiments was carried out to show how retrieval could be improved for mixed-language IR using the proposed approaches, whenever a centralized architecture is used. Moreover, well established baselines, to which the proposed methods would be compared, were also included in this set of experiments so to serve as competitive benchmark runs.

The second set of experiments, which consists of one study that contains four experiments, was set into a traditional distributed architecture. Current traditional distributed architectures are not optimal for indexing mixed and multilingual collections. Accordingly, a newly developed architecture which combines the basic ones (distributed and centralized architectures) with a reasonable re-weighting

component was tested in this chapter. Such architecture should have the ability to efficiently index and retrieve documents, taking into account the mixed-language feature in both queries and documents.
Before delving into the two set of experiments, the chapter presents firstly the common work that was set. Such work includes prior-to-indexing normalization in texts, identified fields for indexing, stemming and translation of queries and their utilized translation resources. These details of the test environment are presented in section 6.1. Section 6.2 is devoted to experiments and results. Both sets of the experiments were presented here. Thus, the proposed methods in the centralized architecture and those in the traditional distributed architecture were investigated and analyzed in this section and they were also compared to different baseline runs. Section 6.3 summarizes the findings.

6.1 Experimental Setup and Test Environment

The test environment of the experiments had been created using version (B) of the MULMIXEAC test collection, which contains textual data only as described in chapter 4. Although, an index for the MULMIXEAC collection was created, indexing documents for experiments was different from their indexing when the data set was analyzed. This is because terms for an IR retrieval process are often stemmed, and thus a new index should be set up. This part is devoted to show the common activities that are often set before indexing. Such activities involve how the mixed and multilingual texts were normalized, what stemmers were utilized, what were the used translation resources, etc.
It important to note that in spite of the fact that two different indexing architectures (centralized and distributed) were utilized in the following experiments, the same *prior-to-indexing* processing steps (i.e. stemming, normalization, etc) were used. Hence, the prior-to-indexing procedures described in the next sub-section are common and they were applied to all experiments, regardless of the architecture of indexing. Furthermore, it is essential that the same steps of processing were also applied to terms in queries, as they were applied to documents.

6.1.1 Prior-to-Indexing Normalization

As texts in all documents of the MULMIXEAC test collection were previously tagged with a language attribute <lang>, which shows the language of a certain word/phrase/portion/ paragraph/document, the process of extracting index terms for normalization was straightforward. For the Arabic texts in both monolingual Arabic the mixed documents, the prior-to-indexing step begins with processing the kasheeda (see section 3.3.1 in the Arabic IR chapter). Diacritical marks were next removed. Following this, a letter normalization process for the Arabic texts (see section 3.3.1 in the Arabic IR chapter) was also executed so as to render some different forms of some letters with a single Unicode representation. The letter normalization that had been performed for Arabic words in documents includes:

- Replacing the letters HAMZA (ٱ،أ) and MADDA (آ) with bare ALIF (ا);
- Altering final un-dotted YAA (ى) with dotted YAA (ي);

- Replacing final TAA MARBOOTA (ة) with HAA (ه); and
- Modifying the sequence ءى with ئ.

English documents and English parts in mixed documents were also normalized in terms of case-folding, which alters all letters into one case (i.e. lower case).

6.1.2 Text Processing and the IR System

Since the attribute <lang> can be assigned either Arabic or English, four logical field types, as in the index that was created for evaluating data, were utilized to populate text during the indexing stage, which is described later. These fields are <*TITLE-Arabic*>, <*CONTENTS-Arabic*>, <*TITLE-English*> and <*CONTENTS-English*>. Depending on both the architecture of indexing and language(s) of documents, some or all fields may be used. But, as a basic assumption, the <TITLE-Arabic> and <CONTENTS-Arabic> fields were used to populate data taken from texts in monolingual Arabic documents or from Arabic parts/snippets/paragraphs in mixed documents, whereas the <TITLE-English> and <CONTENTS-English> fields were applied to monolingual English documents or the English portion(s) in mixed documents. Specific details concerning which fields would be utilized, are provided later, when needed, in the experiment methodologies.

As there are many fields, during indexing, language dependent processing for stemming and stopwords removal was applied. This is *per-field-analyzer-wrapper* in Lucene, which was used in all experiments reported here. Therefore, whenever an indexing architecture is planned (centralized or distributed), Arabic words were lightly stemmed using the LIGHT-10 stemmer, which was demonstrated in section 3.3.3 in the Arabic IR chapter, before they were populated in the <TITLE-Arabic> and <CONTENTS-Arabic> fields, whereas the English words, which were populated in the <TITLE-English> and <CONTENTS-English>, were stemmed by the SNOWBALL stemmer[21]. Both the stemmers are built into the Lucene IR system.

Before populating fields with the appropriate stemmed terms, stopwords were also eliminated. Both Arabic and English stopword lists are built in Lucene. Nevertheless, some stopword entries in the two initial lists were removed. This is because the MULMIXEAC corpus is on common computer science, which makes the removal of a stopword like 'for' may probably a bad decision, although keeping it results in retrieving a lot of documents. Many commercial IR systems may index documents under all available forms and may not apply a stopword list (Savoy, 2007).

All experiments were conducted using the Lucene IR System with some integrated components, developed by Perez-Iglesias et al., (2009). In that integrated component, the developers extended the Lucene to a more advanced ranking model, which is the probabilistic BM-25F and BM-25 to multiple weight fields (see section 2.1.3.2.2) of the Okapi BM weighting (illustrated in formula 2.23).

However, some parts of the code were modified. In particular, since the Kowk's approximation to Pirokola's structured query is central to the design of the work presented here, some of the Lucene

[21] http://www.snowball.tartarus.org/

classes had been extended/modified to apply the proposed cross-lingual Kwok's approximation. The major idea was based on extending the *TermScorer* class. Instaed of using accepting only a term in its constructor, the class has been modified to accept an array of *TermQuery*. Ecah array contains an English term and its translations. Next, inside the methods *Score* and *explain*, the array of the *TermQuery* is iterated to get the TF and and DF for all elements (term plus its translations) as ysnonyms. The modification of the *Scorer* class would have its effect, when it is declared, inside the *BooleanScorer* class to include the required array of *TermQuery*. This in turn would affect both *SingleBooleanScorer* and *BooleanWeight* classes.

The tuning parameters values that were set in the Okapi implementation for experiments presented here were 2 and 0.5 for K_1 and b, respectively.

6.1.3 Queries and their Translations

Topics in the MULMIXEAC collection contain several fields, but the only mixed field among them is the *OriginalQuery*, as it was described in section 5.2.2.4. In the experiments reported here, the OriginalQuery field in all topics was used as a source query. Since, there were two languages presented in each individual mixed query, two directions for translations can be identified, a translations process from Arabic to English and vice-versa. Thus, the overall idea is that for a single mixed source query, English portion is translated to Arabic, whereas the Arabic portion on the same mixed query is translated to English. This would result in a bi-directional translation. The principal behind the term 'bi-directional translation' here is different from its peer in CLIR. In CLIR, the term bi-directional translation (i.e. Boughanem, et al., 2002) refers to the use of a hybrid approach that merges both document translations, from one direction, with query translation, from the other direction with the underlying assumption that precise translations tend to backward translate to the source term. This idea of bi-directional translation is different from the one presented here and it is covered in brief in the next sections.

After each portion in a certain source language is translated, the newly produced portion in the target language is concatenated to the part in the original mixed query, whose language is similar to the language of the produced translated part. This would formulate a monolingual query in a certain language. For example, in the query DLIB01, which was 'مفهوم الـ deadlock', the English translation for the Arabic term was found to be the word 'concept'. Therefore, both the words concept and deadlock, which is the English word in the original mixed query, are attached to each other to form the monolingual English query 'concept deadlock'. Notice that regardless of the translation direction, the translation process was performed after the removal of the stopwords. For the translation, five translations and/or transliteration resources of three types were used. The first type is a special English-to-Arabic computer-based dictionary. This is an in-house-built dictionary, that was compiled from various resources with approximately 17,500 entries, all of them technical terms in common computer science vocabulary and with many regional variants. The size of the dictionary is small, but this was the only available resource at the time of conducting these experiments. Nevertheless, some reported studies, as discussed in section 2.2.3.1 in the review chapter, showed practically that the larger the dictionary size, the greater the effect

on retrieval performance till the size reaches a certain number of entries (size), specifically, 10,000 words(Xu and Weischedel, 2005). This is especially true when the dictionary contains the most frequent words. Accordingly, increasing the total number of entries, does not necessarily would have a positive impact on CLIR effectiveness.

The technical dictionary was also inverted after making a copy, in the second direction (from Arabic-to-English) with some modification to its entries in order to account for those technical terms that may be present in Arabic, if any, in the original mixed queries. However, such terms are very few in the query set and relatively few in Arabic vocabulary, in general.

The second type of resource was the transliteration probability table of AbdulJaleel and Larkey (see section 3.5.2 in the Arabic IR review), which is provided from English-to-Arabic and was trained by the same developers on a parallel list of 125,000 entries containing person names and place names in both Arabic and English. This transliteration table was used, when needed, for the transliterating English OOV words into Arabic.

The third type of translation is a Web-based statistical machine translation system that is the Google translate[22]. This source was used to translate Arabic words, which are merely taken from general purpose vocabulary, into English with at most one sense returned. The next sections show how these resources were employed for translating source mixed queries.

6.1.3.1 Translation of English Portions

English portions in mixed queries are assumed to be technical terms. Thus, for their translation a sequential translation approach (see section 2.2.3.1 in the review chapter) was utilized. Accordingly, for each term in the English portion its translations were looked up in the technical dictionary word-by-word. If there is more than one translation present, all of them are retained and used. This is necessary in scientific domains, especially in the Arabic language due to the large number of regional variants. If a certain term is not covered in the computer-based dictionary, the successive techniques of backoff translation, illustrated in section 2.2.2.3, were utilized.

If the term resulted in an OOV case, it would be both transliterated, using the hand-crafted statistical transliteration model of AbdulJaleel and Larkey, and translated, using the Google translator. Among all the possible transliterations that were produced using the transliteration model for a specific OOV term, the top 2 were selected as transliterated terms for that technical English term. The value 2 was selected with human judgments experimentations. In particular, lists of different English technical terms were used to produce the Arabic equivalent transliterations. Next, the obtained Arabic transliterations were analyzed by experts in computer science, who concluded that the top two transliterations on average are valid and/or may appear in Arabic text. For the translation only the first sense was retained. This is to avoid flooding the translated portion of a query with many common words since the corpus is specialized and the Arabic language is rich with synonymous words.

[22] http://translate.google.com

During the entire steps of translation and/or transliteration, the information of whether a certain translation of a certain term was obtained from the technical dictionary or not, was stored in a temporary data structure. This is important for the measuring criterion, of whether weights of terms should be re-weighted or the original weights kept, as discussed before in the design chapter.

The use of both translation and transliteration approaches together is different from CLIR because in the latter usually terms are transliterated only when their translations in a target language were failed. The rationale behind performing both translation and transliteration of English term, when not found in the technical dictionary was twofold. On one hand, many computer-based technical terms in Arabic are translated rather than transliterated, e.g., ذكاء اصطناعي (meaning: Artificial Intelligence). On the other hand, many of them are just transliterated rather than translated, such as in بروتوكول (meaning: Protocol). Such family of terms includes language names (Pascal), components (java beans), hardware and peripheral names (mouse), general purpose mathematical functions (power), etc. Thus, performing translation and transliteration of words together usually results in a non-empty set. It was possible to apply a more robust technique before attempting to translate and transliterate the English OOV term. In particular, the optimal solution would probably be the submission of these technical terms to a search engine, using its APIs, and to make use of co-occurrence measures so as to extract possible translations, as discussed in the section 2.2.4.1.1. Nevertheless, utilization of such approaches depends on the availability of a search engine API key, which is not applicable in some Arabic countries due to political considerations. Thus, the simple method of both translation and transliteration was adopted.

As a result, for these consecutive steps of sequential translation, the English portion was being translated into Arabic. However, the next logical step is to merge the obtained translated words, which are in Arabic, with the portion that appears in the same language (Arabic) in the corresponding original mixed query, and thus a monolingual Arabic query is obtained.

6.1.3.2 Translation of Arabic Portions

With respect to the Arabic snippets in mixed queries, they are mostly taken from general-purpose vocabulary. Nevertheless, there were some cases in the query set, although there were very few, in which the technical term was written in Arabic, rather than in English. Therefore, the same methodology of translating the English portion was applied. Thus, if a technical Arabic word appears in the Arabic portion in a mixed query, its translation is obtained from the special dictionary. However, the English terms in mixed queries were presumed to be technical. Contrary to this case, technical Arabic terms cannot be identified in mixed queries, therefore a matching process was carried out for all terms in the Arabic portions with the technical dictionary and with the same methodology of backoff translation, when needed. If the translation of the Arabic word from the technical dictionary fails, then the word is probably taken from general vocabulary. Such a word was translated using the Google translator with a token-token mapping approach. In this dominated approach of query translation in CLIR (Nie, 2010, Ture, et al., 2012) each word is translated individually and again only the top sense is considered. Thus, given a source-langauge term in Arabic presents in a query *a*, its first translation in the target-language

(English) *e* is obtained using Google translate and is considered as its sense. This means that all alternative translations are discarded. The process of this translation is performed using a simple written application that is based on some opened Google APIs. As in the translation of the English portions, the information about the source of translation, and whether it is the technical dictionary or not, is stored for restricted re-weighting criteria. When the translated English part is concatenated to the original English portion in the corresponding mixed query, a monolingual English query from the source mixed one is produced. The stage of translating a mixed query eventually resulted in two monolingual Arabic and English queries plus the original mixed query. These queries, mostly the monolingual versions, were used in experiments.

6.2 Experiments and Results

This section reports results of the experiments that were conducted to test effectiveness of the techniques shown in the design chapter. As previously described in that chapter, the problem of mixed-language queries and documents has been approached mainly as a re-weighting scheme component, the classical problem of the CLIR, and/or as an indexing architecture component. Hence, these two components (either both or one of them) were developed in a centralized or a distributed architecture. Accordingly, there are two different sets of experiments. The first set utilized a centralized architecture of indexing, while the second set made use of a traditional distributed indexing. The next sections report results in each for these two sets.

6.2.1 Experiments of Mixed-Languages in a Centralized Index

This section describes the experiments that were carried out to evaluate the developed techniques in a centralized environment for indexing and retrieval. It also compares these techniques to three different baselines. Therefore, this section contains four studies. It begins with a study, named as study I, consisting of three different types of baseline experiments that were conducted together for comparison reasons. These three experiments would serve as baselines to which the developed techniques would be compared. In particular, the three conducted baselines were 1) an upper baseline that was based on a monolingual ranking and retrieval using manually translated English queries. The utilized index in this run was a centralized index, in which all documents are placed together 2) a lower baseline that combines both the structured query model(s) for estimating weights, after queries were translated, with the centralized approach of indexing and 3) a search-engine-retrieval-like baseline, which simulated how existing search engines handle mixed queries.

The next set of experiments in a centralized index was called study II and it was devoted to evaluate the effectiveness of the cross-lingual structured query model, described in section 4.2.2. In particular, three runs were conducted in this study to test the effectiveness of the proposed components.

The third study, named as study III, tested the performance of using a weighted inverse document frequency, described in section 4.2.3, in terms of a sub-collection damping factor incorporated in the

computation of terms' weights. The study showed that such a use of a reasonably weighted inverse document frequency could have a significant impact on mixed-language retrieval.

Finally, study IV tested the effectiveness of the combination of the two approaches (cross-lingual structured query model and weighted inverse document frequency).

As all experiments in this part make use of the same centralized index, there are some common methodologies that were used by all the four studies.

Common Methodology

In experiments reported in this part, from study I to study IV, all documents in their textual forms had been put into one index pool. As the collection contains both mixed and monolingual documents, all the previously recognized fields (<TITLE-English>, <CONTENTS-English>, <TITLE-Arabic> and <CONTENTS-Arabic>) for indexing documents were used, as discussed in section 6.1.2. Terms were firstly normalized, stemmed and stripped-off, if they are stopwords, according to their matching stemmer.

In section 6.1.3, which was the translation process, it was shown that two equivalent monolingual queries (one in Arabic and the second in English) would be obtained for each bi-directionally mixed query. Because a centralized architecture was utilized, these two monolingual versions were merged to form another big and yet mixed query, in the two languages. For example, in the previous mixed query, which was 'مفهوم الـ deadlock', the big concatenated query would likely be 'الجمود التوقف التام الإغلاق الإقفال مفهوم الاستعصاء concept deadlock', in which words presented in the same colour are translations of each other. Queries that were automatically generated in this way were submitted one by one to the previously created heterogeneous index, after applying one of the proposed re-weightings, which is dependent on the run. Such mixed and big queries, which will be referred to as *mixed merged queries* through this chapter, were used for document retrieval of experiments in study II, study III and study IV. For study I, the same approach was used in the lower baseline. For the upper baseline, which was essentially a monolingual experiment, only the monolingual English query set was used. In the search-engine-retrieval-like baseline the original mixed query, rather than the merged one, was used, as this run was conducted to evaluate similar retrieval of current search engines.

The graded relevance was used for estimating retrieved document relevance levels, after utilizing a pooled assessment, as described in sections 5.2.3 and 2.4.2. The Discount Cumulative Gain (DCG) was used to measure performance and it was computed for the top 10 documents for each query in the query set used for retrieval, as the retrieval task emphasizes highly relevant documents. The DCG values across all the 47 queries were averaged and the statistical Student's t-test measure was used to compare significance of differences among the conducted experiments.

6.2.1.1 Study I: Baselines

Experiments in study I were carried out to report baseline runs, to which all experiments in this chapter would be compared in terms of retrieval effectiveness. In particular, study I contains three baseline experiments, each of which functions for a specific baseline task:

- The first experiment, which was called $\mathbf{b_{IR}}$, is a monolingual English run, in which queries were translated manually by human experts. The b_{IR} experiment was conducted to upper-bound retrieval effectiveness for CLIR experiments reported in this chapter, as its retrieval was considered as an unreachable upper-bound, beside the well known measures (i.e. DCG) for effectiveness.
- The second experiment was a CLIR baseline that combines the centralized approach of indexing with the structured query model(s) for weighting. This experiment should determine the impact of using mixed-language queries in current CLIR weighting and retrieval. Furthermore, the experiment should determine if the effectiveness of current weighting and indexing would differ when moving from news genre to specialized computer-based domain. The experiment developed to handle this approach was called $\mathbf{b_{CLIR}}$. This is the lower baseline.
- The idea behind the third experiment, which was called $\mathbf{b_{IRengine}}$, arose from searching capabilities of existing search engines and how they handle mixed queries. Thus, the $b_{IRengine}$ experiment mimics, and thereby exploits, retrieval of search engines (search-engine-retrieval-like), in which mixed queries are posted to the MULMIXEAC collection as they were submitted by originators, who were typical Web users. Obviously, this experiment can be considered as another reference point, because it is derived from current nature of realistic Web, to which the performance of proposed CLIR approaches would be compared.

Methodology

As there were three experiments to be conducted together in study I, there were also three different methodologies.

a) Upper Monolingual Baseline (b_{IR})

In CLIR one common technique for evaluation is to compare findings produced by a certain new technique to those obtained from a monolingual retrieval. Such evaluation is often assessed in terms of percentages computed for both the proposed technique and the monolingual retrieval. Queries for such a monolingual retrieval run are often translated manually to a target monolingual language by human experts, as the automatic translation usually may introduce noise.

In mixed-language queries and documents, it is possible to conduct two different upper baselines, one in English and the second in Arabic. This is because the test collection is mixed and multilingual in these

two languages. But, it is not surprising that the number of the Arabic documents in the test collection is very small, compared to the English sub-collection size, even on the Web as shown in the introduction. Furthermore, English documents are much richer, compared to those in Arabic. Accordingly, the upper monolingual baseline was conducted with manually translated English queries. The translation task was performed by professionals, who are staff members in computer science, and it produced a set of 47 monolingual English queries. The Arabic portions in the original mixed queries were the only parts to be translated.

In CLIR, it is often that the document collection is in one language. Thus, whenever an upper baseline experiment is conducted, the process becomes of monolingual task. In experiments reported in this chapter, the document collection is mixed and multilingual, and thus, in the upper baseline, all documents in the MULMIXEAC were used, instead of utilizing monolingual English documents only. Removal of mixed and monolingual Arabic documents for conducting monolingual runs, as in the upper baseline, would result in using two different document collections (one for the monolingual run and the other for the experiments). To avoid such unfair comparison all documents in the MULMIXEAC collection were used in the b_{IR} experiment. This is essential in IR.

Thus, in the methodology implementation in this experiment, each manually translated query in English was posted to the centralized index of the entire MULMIXEAC collection and the top 10 documents were retrieved in order to compute the DCG for evaluating effectiveness. The average DCG for all queries was then considered as an upper ceiling for effectiveness evaluation and, thus, it was utilized for comparison purposes.

b) Cross-lingual Lower Baseline (b_{CLIR})

There is not much previous work on the mixed-language problem of queries and documents as the dominant belief is that a CLIR task is a translation process followed by a monolingual retrieval, which results from the use of translated monolingual queries. It was also shown that the mixed-language problem in queries and documents can be considered as a compromise approach that falls between both the CLIR and the conventional MLIR approaches. From this perspective, the b_{CLIR} is also a combined run between some well known and tested methods in both these two fields, that is, a centralized index merged with a structured query model.

On one hand, the centralized approach of indexing, which is the most dominant architecture on most CLIR approaches, is a well known approach and a widely reported baseline in traditional MLIR. Furthermore, the centralized architecture has some major similarities with the work presented here. Firstly, its document collection is often multilingual in various languages. Secondly, its querying strategy is based on mixed queries, because the source query, which is almost monolingual, and its translation, which is monolingual too, are concatenated to each other to formulate a big mixed and merged query. These two major similarities make the centralized index the closest approach to the work reported here.

On the other hand, structured query model approaches are widely reported baselines in CLIR. The process of structuring queries contributes to translation disambiguation as it groups together all the

alternative translations (in a target language) of a source query term in a certain source language, and thus structuring results in the inclusion of translation disambiguation during the retrieval process. In that context, the lower baseline is a comporimise method that combines both the centralized architecture and the structured query model. For example, in the previous mixed query, which was 'مفهوم الـ deadlock', the centralized architecture will firstly translate each word in this query bi-directionaly and then the generated queries will be concatenated. This would result in the big query 'الجمود التوقف التام الإغلاق الإقفال مفهوم الاستعصاء concept deadlock', in which words presented in the same colour are translations of each other. Next, the stutured query model would cluster all the Arabic translations of the the English technical terms together monolingualy rather then cross-lingually, and thus, the Arabic translations would be synonyms to each others. In SMART notation this is typically to using the SYN operator, e.g., #SYN(الجمود الاستعصاء التوقف التام الإغلاق الإقفال). In that context, the use of the SQ models for structuring Arabic translations of English technical terms in the lower baseline could ensure that all the alternative translations (regional varinations) of English technical terms in the translated versions of queries, will be included in the final expanded query as synonyms. Note that Arabic translations are regionally variants and thus the use of structuring mechanisms is important. Thus, the final big query can be represented in SMART notation as:

#SUM(concept deadlock # SYN(الجمود الاستعصاء التوقف التام الإغلاق الإقفال مفهوم)

Although several variants of structured query models were proposed (i.e. probabilistic cross-lingual structured query model), the lower baseline utilized the introduced Kwok version of the structured query model (equation 2.30 for estimating TF and equation 2.33 for estimating DF). This is mainly because of the reasons that were discussed in section 4.2.5. Thus, using a combined approach that merged both the centralized architecture of MLIR with the structured query model of Kwok to represent a lower baseline seems appropriate and a well established method for the case of mixed-language queries and documents. Thus, in the methodology of the b_{CLIR} run, the monolingual versions of each source mixed query, which were obtained as mentioned before, were merged together to form another mixed query, but all the Arabic alternatives translations, which were obtained from the technical dictionary, of each English technical term were considered as a unique term (synonyms in the Arabic language), as described above. Thus, the term frequency and document frequency components during weight computation for these translations were estimated according to equation 2.30 and 2.33, respectively, in the review chapter. Queries that were obtained in this way were submitted one by one to the big mixed and multilingual index.

c) Search-Engine-Retrieval-Like Baseline ($b_{IRengine}$)

In realistic Web environments, mixed queries are often posted to search engines with no translation mechanisms offered, although machine translation software is usually advocated to translate retrieved pages. In that context, it is reasonable to establish another mixed-query run as an alternative baseline, in which the original mixed queries are employed. This mixed-query run is neither a CLIR run, as it eliminates the use of any translation mechanism, nor a monolingual run, since its queries are originally

mixed. It is used as a realistic Web-based reference point that reflects the impact of submitting mixed queries to search-engine-like systems. Hence, in this $b_{IRengine}$ run, queries were submitted as they were posted by original creators to the single mixed and multilingual index.

Through these experiments, both the lower baseline and the search-engine-retrieval-like runs are referred to as the mixed-query baselines, whenever comparisons referred to both of them together.

Results and Discussion

The following table, Table 6.1, shows the performances obtained in each of the three baselines' retrieval. Values in the table are presented in an average DCG at document cut-off levels from 1 to 10 and the first top 10 documents retrieved are used for the final performance evaluation. The use of all points (from 1..10) are provided for drawing the curves for each run. Figure 6.1 plots the results of these three baseline runs together in a single graph.

Measure	Average DCG @									
Run	1	2	3	4	5	6	7	8	9	10
b_{IR}	4.447	8.745	11.403	13.424	15.174	16.771	18.211	19.409	20.671	21.882
b_{CLIR}	3.340	6.277	8.612	10.357	11.640	12.759	13.904	14.783	15.770	16.641
$b_{IRengine}$	3.040	5.250	7.539	9.249	10.255	11.298	12.165	13.125	14.335	15.064

TABLE 6.1: Results of different baselines. The upper baseline (b_{IR}) is a monolingual run, the lower baseline (b_{CLIR}) is a CLIR run and the search-engine-like baseline ($b_{IRengine}$) is to mimic search engine's retrieval. Values are average DCGs taken for 47 queries over a single index of the MULMIXEAC test collection.

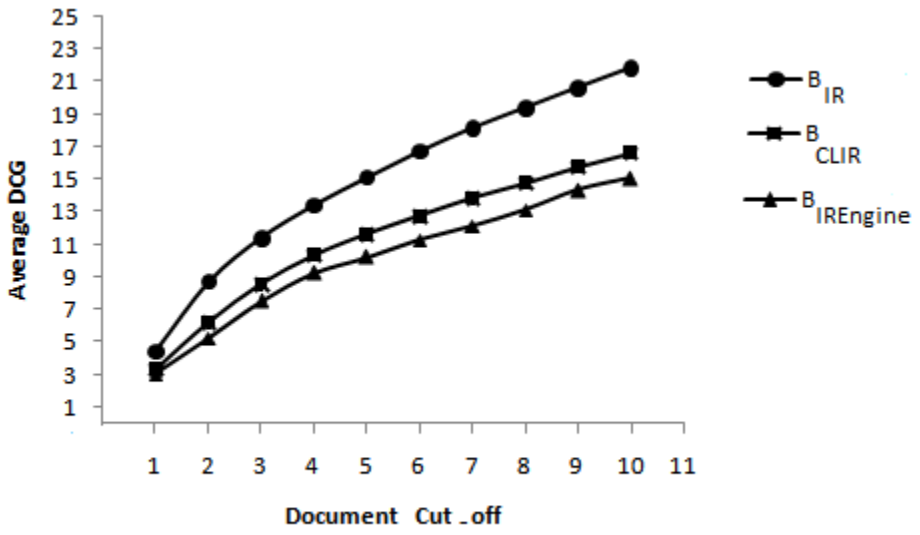


FIG. 6.1: The average DCG curves at document cut-off values[1..10] for the monolingual upper baseline (b_{IR}), the CLIR lower baseline (b_{CLIR}), which comprised the centralized architecture with the Kwok formula of SQM, and the search-engine-like baseline ($b_{IRengine}$).

In the figure, the best results are related to the upper baseline run (b_{IR}). This was not unexpected, as it was estimated that the upper baseline would significantly outperform not only the naive search-engine-retrieval-like baseline ($b_{IRengine}$), but also the more sophisticated lower baseline run (b_{CLIR}). This is because the retrieval of the upper baseline was performed using manually translated queries with experts and, hence, no noisy translations could affect retrieval. Furthermore, as the results of the monolingual retrieval were obtained using the English language, whose documents are often much richer in their contents, compared to those in Arabic and/or bilingually mixed documents, the performance is unsurprisingly very good.

It can also be seen in the figure below that the retrieval efficiency for the lower baseline (b_{CLIR}) run goes down and reaches a declining percentage of approximately 24% (average DCG at rank 10 was 16.641, whereas it was 21.882 for the b_{CLIR} run), compared to the full efficiency of the upper baseline run at top 10 ranked documents. However, the difference begins with small values (at rank 1 and 2) since both mixed-query runs also retrieved good documents on these ranks. But, as documents were accumulated, the difference in effectiveness became relatively higher. A similar drop in the effectiveness of the naive search-engine-retrieval-like baseline ($b_{IRengine}$) also occurred. In particular, at rank position 10, the performance of $b_{IRengine}$ run falls to a low minimum of %31, compared to upper baseline and %10, compared to lower baseline. This decline in performance of the mixed-query runs (b_{CLIR} and $b_{IRengine}$), was mainly caused by the fact that these runs were often attempting to perform exact matching between queries and documents, but with no sufficient analysis of the type of the submitted query (monolingual or mixed) and regardless of the language presented in each. Furthermore, the overweighting problem contributed to bad performance of the mixed-query runs. Particularly, it causes many terms, mostly in Arabic, in the mixed queries (original or mixed and merged) to overweight, as their corresponding sub-collection/language size, typically the Arabic one, included in the big multilingual collection is small. The consequent result for this over-weighting problem in mixed-query baselines, as well as the exacting matching problem, was that their retrieval lists were dominated by mixed documents. It is likely that this bias towards mixed documents in these two mixed-query runs is the artifact of existing approaches, which do not consider how much a query is mixed, due to underlying assumption of monolingual weighting and retrieval.

Contrary to these drawbacks, the upper baseline retrieval was not affected by such overweighting problems, as its queries were essentially monolingual in English. Consequently, the majority of the top retrieved documents, and mostly the majority of the retrieved documents, by this upper baseline run were monolingual in English. It appears that using monolingual English queries with a multilingual document collection would definitely cause non-retrieval of Arabic monolingual documents, whereas English documents rather than those mixed ones would dominate the entire list because their scores are predicted to be higher than those mixed, due to their partial matching for only parts of the monolingual English queries.

When a point-by-point comparison is considered in Figure 6.1, it can be observed that English documents are much richer in their content, compared to mixed and monolingual Arabic documents. This is obvious if the average DCGs of the first documents, for example, are compared (i.e. the first one or two document

cut-off values). In spite of the fact that the retrievals of the mixed-query baselines attempt to find the relevant information, they resulted in lower average DCG values. This result mainly confirms that English documents are much richer in their content, compared to mixed and monolingual Arabic documents.

Comparing the b_{CLIR} run to the $b_{IRengine}$ baseline in the figure, document retrieval with the latter run was consistently the worst retrieval effectiveness, although it presents what could be achieved currently with existing search engines. The difference in retrieval scores is statistically significant (p-value < 0.000052) at top 10 documents, for the b_{CLIR} run, using the Student t-test significance measure. This difference can be explained from different view. First of all, the length of the queries in each corresponding query set for the two runs were different. In particular, in the b_{CLIR} baseline, all the alternative translations, mostly in Arabic, for English technical terms were included while they were not in the $b_{IRengine}$ run (i.e. 'ال مفهوم deadlock'). The absence of all the candidate translations in the original mixed queries in the $b_{IRengine}$ run makes the query insufficient for good retrieval as they are relatively short and, thus, the useful clues are few. The results revealed that in the $b_{IRengine}$ run more weights were given to source non-technical terms, mostly in Arabic, in the original mixed queries than weights of those English technical terms. This bias towards non-technical Arabic terms was caused by the over-weighting problem, in which such non-technical terms tend to have lower document frequencies (and thus relatively high weights are assigned), because the Arabic collection size is small, whereas the technical terms, whose languages are mainly in English, would probably have relatively high document frequency. This is an undesirable characteristic, as the weight of non-technical terms would artificially inflate. In the b_{CLIR} run the presence of both the technical term with its translation(s) (i.e. deadlock and الإقفال) also causes the Arabic terms to overweight, as the merged query is mixed, but the impact was moderate, due to the length of the query and the inclusion of these translations. Furthermore, the structured query model, which clusters all the Arabic translations of a single technical term, in the b_{CLIR} run diminished the impact of individual computations of these translations and contributes to the retrieval performance of the b_{CLIR} baseline.

Nevertheless, examination of retrieved lists in the two mixed-based query runs revealed also an interesting anomaly, which was a credit to the $b_{IRengine}$, rather than to the b_{CLIR} run. In some cases the retrieval performance of a few mixed queries in the $b_{IRengine}$ run was somewhat similar to the b_{IR} run, exceeding the effectiveness of b_{CLIR}. This anomaly result stemmed from the occurrence of only Arabic stopwords, instead of words from general-purpose vocabulary, in the original mixed queries. For instance, in a query like ' ما هو Mutual Exclusion' (meaning: what is mutual exclusion), the Arabic portion 'ما هو' is entirely composed of stopwords. Thus, during the query processing phase only technical terms, which are in English, are preserved. This was the main reason for the somewhat better performance of $b_{IRengine}$.

6.2.1.2 Study II: Cross-lingual Structured Query Model

In the proposed cross-lingual structured query model, which re-estimates term frequency, document frequency and document length components, documents are re-weighted and re-ranked according to their relevance scores. Experiments in study II were thereby carried out to evaluate retrieval performance

of the proposed cross-lingual structured model. The evaluation of the proposed r-weighting scheme was investigated in this study with two experiments, each of which was utilized to test a specific part of the proposed cross-lingual re-weighting:

- The first experiment, which was called CRSQM-NODECAY (stands for CRoss-lingual Structured Query Model with NO DECAY), attempted to investigate the effectiveness of estimating both the term frequency and the document frequency components crosslingually. But, the neighboring feature of Arabic terms that tend to co-occur with their equivalent English translation(s) (i.e. as in 'deadlock الأقفال') was not considered in the weight estimation.
- The second experiment tested the proposed re-weighting scheme after using a damping factor for bilingual paired terms that tend to co-occur together. The experiment was called CRSQM-DECAY (stands for CRoss-lingual Structured Query Model with DECAY).

The next sections detail these two experiments.

Aims

The main objective of study II was to describe the impact of using synonymy across languages (cross-lingual structured query model) on weights of documents. It also showed how the co-occurrences of bilingual terms in documents could affect documents rankings significantly. Furthermore, the study compared findings of the proposed cross-lingual structuring approach to that of the baseline runs. The following research questions were posed:

1. Whenever synonymy across languages (cross-lingual structured terms) is considered, what are the beneficial impacts on retrieval effectiveness of mixed-language querying? Does the use of such an approach will result in breaking dominance of mixed-language documents, especially on top of retrieved list? The hypothesis here is that neither a damping factor for term frequency of co-occurring bilingual terms nor document length reduction are being considered. This is the CRSQM-NODECAY run.
2. Does the co-occurrence of Arabic technical terms with their English equivalents affect rankings of documents, especially those are highly relevant? Furthermore, does the proposed re-weighting scheme for handling such co-occurrence of bilingual pair terms result in an improvement over the cross-lingual re-weighting in the CRSQM-NODECAY run? Does such method minimize the extra weights that were earned by bilingual co-occurrence of terms? Tackling the term frequency component of bilingual co-occurred terms and making use of a damping factor for it in bilingual terms is the CRSQM-DECAY run.

Methodology

The key idea of improving performance in study II is to modify weights of terms, in the merged mixed queries produced in section 6.1.3. However, in all experiments of this study, a weight of a term is modified if and only if its translation(s) is obtained from the technical compute-based dictionary. Hence,

in the methodology of the first run, which was the cross-lingual structured query model with no decaying factor for bilingual co-occurring terms (CRSQM-NODECAY), a cross-lingual structuring mechanism was applied for each term in the mixed merged query, whose translation criterion of producing senses from technical dictionary holds. Such structuring makes use of equation 4.1 for estimating the term frequency component. The same cross-lingual estimation was also applied to the document frequency component according to equation 4.6, which was the proposed cross-lingual version of the Kwok formula for document frequency estimation. For example, if the mixed merged query ' الجمود التوقف التام الإغلاق الإقفال مفهوم الاستعصاء concept deadlock', which was previously shown in the common methodology section, is submitted, then the equivalent translation words of the individual source term 'deadlock' and the source term itself, are cross-lingually structured. This would handle every translation of the source term (i.e. الجمود) and the source term itself (i.e. deadlock') as instances of that source term and, thus, a set of cross-lingual synonyms that contain the terms 'الجمود الاستعصاء التوقف التام الإغلاق الإقفال deadlock' is created. The net effect is primarily occurred in TF and DF computations. For example, in term frequency, number of occurrences of the source term would be handled as the sum of the number of occurrences of all term in the synonym set, which contains terms in two languages, but regardless of languages of these terms. Thus, the entire process is repeated for every term, whose translations are obtained from the technical dictionary, in the mixed merged query. Next, weights of all terms are added to produce the document relevance score. Note that weights of terms, whose translations were not produced by the technical dictionary, are kept with no modifications.

Processing of co-occurring bilingual terms in terms of a decaying factor for term frequency component was applied in the CRSQM-DECAY run. The methodology of this run uses firstly the same procedures as those presented in the previous CRSQM-NODECAY run. In particular, terms in the crosslingual synonyms set of a source term is firstly manipulated, in terms of TF and DF, as instances of that source, as described above. Following this, an additional weight modification, which is the use of the decaying factor as in equation 4.3, is applied. The decaying factor reduces the term frequency of source term if it occurs with one of its translation, e.g., 'deadlock الإقفال', which is very common in writing style of non-English documents. Note that the tuning parameter value of K_1 in the okapi BM25, which was the IR model used in all experiments, was set to the value of 2, and it is known that large values for K_1 means documents will be weighted using the raw term frequency. As previously described in design chapter, the bilingual terms across languages are considered as 'co-occurred terms' if they appear together in a window of size 5 and without considering their order of terms (which term appears first and which one appears second).

It was shown that the damping factor of bilingual terms affects the document length component since any reduction in the frequency of a certain term in a document is directly proportional to the number of tokens in that documents. Thus, in the CRSQM-DECAY, equation 4.4, which re-estimates document length, was also applied, whenever damping factor computations were implemented.

Results and Discussion

Measure	Average DCG @									
Run	1	2	3	4	5	6	7	8	9	10
CRSQM-NODECAY	3.319	7.277	9.612	11.559	12.961	14.380	15.598	16.868	18.089	19.249
CRSQM-DECAY	3.915	8.020	10.551	12.540	14.217	15.731	16.959	18.292	19.561	20.746
b_{CLIR}	3.340	6.277	8.612	10.357	11.640	12.759	13.904	14.783	15.770	16.641
$b_{IRengine}$	3.040	5.250	7.539	9.249	10.255	11.298	12.165	13.125	14.335	15.064

TABLE 6.2: Shows the results of the proposed cross-lingual structured query model, compared to the lower baseline (b_{CLIR}) and the search-engine-like ($b_{IRengine}$) runs, in terms of average DCGs computed at document cut-off values [1..10] for 47 queries. CRSQM-NODECAY: Cross-lingual-lingual Structured Query Model with no Decaying factor. CRSQM-DECAY: Cross-lingual Structured Query Model with a Decaying factor for co-occurred bilingual terms.

Table 6.2 reports retrieval effectiveness of the experiments in study II, presented in a tabular form. The results were also compared to those mixed-query baselines (b_{CLIR} and $b_{IRengine}$) produced in study I. Figures 6.2 and 6.3 on the next page depict retrieval performances for the CRSQM-NODECAY and the CRSQM-DECAY approaches, respectively, compared to mixed-query baselines (b_{CLIR} and $b_{IRengine}$). They are

separated into two figures for comparison purposes. In each figure the curve of each approach was compared to the two mixed-query based baselines.

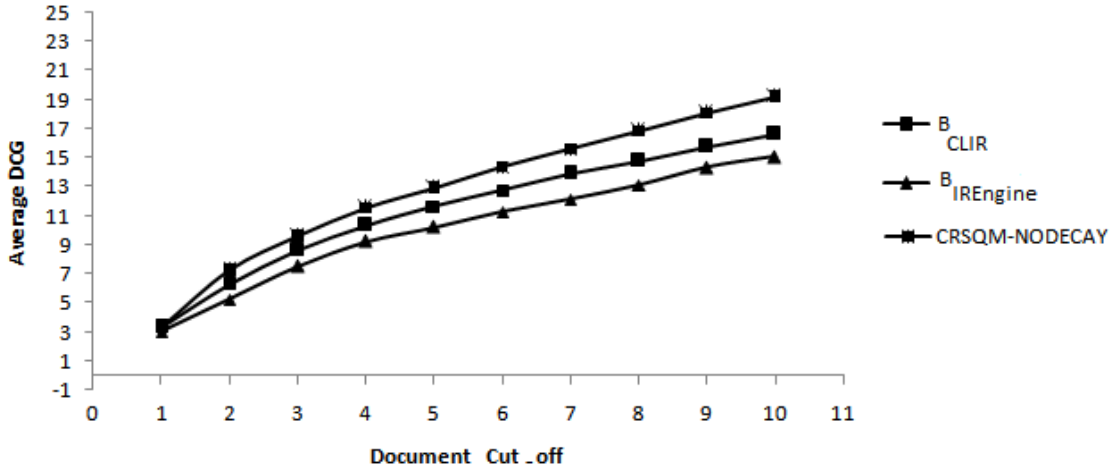


FIG. 6.2: Average DCG curves at document cut-off values[1..10] for the proposed cross-lingual structured query model, but without considering weights of co-occurring bilingual terms (CRSQM-NODECAY run).Curves were compared also to mixed-query baselines(b_{CLIR} and $b_{IRengine}$).

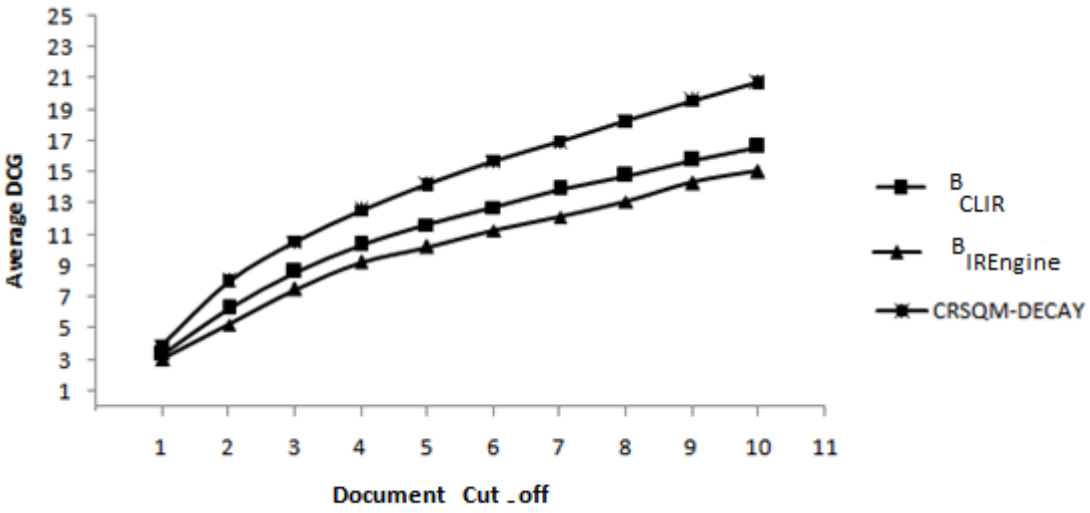


FIG. 6.3: Average DCG curves at document cut-off values[1..10] for the proposed cross-lingual structured query model, when a damping weight factor for co-occurring bilingual terms is considered (CRSQM-DECAY run). Curves were compared also to the mixed-query baselines (b_{CLIR} and $b_{IRengine}$).

The retrieval effectiveness was assessed by the average DCG over 10 points (@ top k documents, k=1..10). As can be seen in the figures, approaches that make use of the proposed weighting algorithms produced more effective results that were consistently higher than both mixed-query baselines (lower and search-engine-retrieval-like baselines).

The performance is statistically and significantly better (using a paired one-tailed, and also two-tailed, Student's t-test with $p<0.05$) for proposed weightings compared to baselines. Table 6.3 lists the *p*-values of significance tests of both the CRSQM-NODECAY and CRSQM-DECAY runs, compared to the lower

baseline b_{CLIR} at document cut-off levels: 2, 3, 4, 6, 8 and 10. Grey cells in the table indicate statistically and significantly better results than the baseline with * to show that $p < 0.05$ and ** for $p < 0.01$, while white cells indicate that there is a difference, but it is statistically insignificant.

Measure	Average DCG @					
Run	2	3	4	6	8	10
CRSQM-NODECAY	0.513	0.192	0.064	**0.007** **	**0.000** **	**0.000** **
CRSQM-DECAY	0.297	0.079	**0.020** *	**0.002** **	**0.000** **	**0.000** **

TABLE 6.3: P-values using the Student's t-test of both the CRSQM-NODECAY and CRSQM-DECAY runs against lower baseline (b_{CLIR}). P-values were computed for average DCG @ (2, 3, 4, 6, 8 and 10).

It is seen in Table 6.2 that the difference in effectiveness of the two runs (CRSQM-NODECAY and CRSQM-DECAY), compared to the two mixed-query baselines, begins with small values at the 1st top document (when k=1) and increases gradually as more documents were accumulated. Although experiments emphasize the task of highly relevant documents, exploring the remainder of each retrieved list (rankings of documents after the first 10) showed that the retrieval performance of the proposed schemes of re-weighting was still much better than that of the mixed-query baselines.

This improvement in performance for the two different re-weighting schemes was attributed to the use of the proposed cross-lingual structured model, in which technical terms in queries are cross-lingually structured, regardless of their languages (by handling them as synonyms across languages and as if they are in a single language). This cross-lingual structuring resulted in that the weight of each technical source term, mostly in English, was calculated as a single weight consisting of re-estimating both the term frequency and the document frequency of the same source term with those in its all cross-lingual synonymous terms and regardless of their languages. This cross-lingual computation in both CRSQM-NODECAY and CRSQM-DECAY runs resulted in different impacts, that were based on text language, on documents (mixed versus monolingual). While in mixed documents structuring technical terms cross-lingually reduces their estimated scores significantly, it reduces, also the scores of monolingual documents, but with a slower rate. Such different impacts on documents stemmed from the different effects of the cross-lingual structuring on English term weights versus Arabic term weights, which in turn were reflected as different effects on the scores of mixed documents versus monolingual documents. It was shown in the discussion of study I that Arabic terms were over-weighted in mixed-query baselines, due to the low number of documents in their corresponding sub-collection. But, when technical terms were appropriately and cross-lingually structured in the CRSQM-DECAY and the CRSQM-NODECAY runs, the document frequencies of Arabic technical terms, which are essentially technical translations, would increase significantly (note that the document frequency of the English technical term, which was relatively high, was added and the collection is dominated by English). Such increase in document frequencies of Arabic technical terms would probably have a reduction effect on their weights and moderates the overweighting problem. As a result, mixed documents, which mainly obtained their higher

scores from these over-weighted Arabic translated terms, may re-weighted into lower weights, depending on their cross-lingual TF and DF statistics. Likewise, document frequencies for English technical terms, instead of Arabic terms, using the cross-lingual structuring, were also reduced, as structuring query terms across languages causes such English terms to expand their weight computations to include their synonymous terms in the Arabic language. But, this increase in the document frequencies of English terms was small because the Arabic sub-collection size was small. Thus, their weights were not affected too much and consequently the scores of the monolingual English documents were not reduced too much. In this way, the overweighting problem in the CRSQM-DECAY and the CRSQM-NODECAY runs was moderated. Hence, the IDF factor of the cross-lingual structuring was used to make a difference in the weights of Arabic terms (mixed documents mainly) versus English terms (monolingual English documents). For instance, the document frequency for the technical English term 'deadlock' in the MULMIXEAC test collection was found to be 1343 (resulting in an inverse document frequency =log(70,000/1343) = 1.717), where as the number of documents in which one/some of its Arabic translations alternatives (they were monolingually structured in the lower baseline run) occur was found to be 297 (and thus IDF =log(70,000/297)=2.372), which is relatively high (approximately 1.4 times the inverse document frequency of the English source term 'deadlock'). When structuring terms cross-lingually, as in the two runs of the proposed re-weighting schemes, this would have the effect of reducing the over-weighted Arabic translations, (the IDF would become log(70,000/1640)= 1.63). The original English term weight was also affected, but the impact is relatively low. The result of this re-weighting in the both CRSQM-DECAY and CRSQM-NODECAY runs was that many monolingual English documents, which were mostly more relevant, would probably be ranked ahead of mixed documents and thus, the dominance of mixed documents on top was broken, although some of these mixed documents were still placed at higher ranks due to their high term frequencies.

The term frequency component in the cross-lingual structuring was another reason for the better performance of the proposed re-weighting schemes in Figures 6.2 and 6.3. This is because structuring technical terms cross-lingually makes the term frequency component in mixed documents versus monolingual documents more comparable because the term frequency of terms would be counted regardless of their languages. When no cross-lingual structuring is utilized (i.e. only monolingual structuring for candidate translations), result list is expected to bias towards the term with higher number of occurrences (either the source term in the big merged query or candidate translations, which are treated as instances, of that source term). The use of the cross-lingual structuring for term frequency suppresses such an impact and consequently causes an improvement in the proposed re-weighting scheme runs.

Contrary to such cross-lingual computations for estimating term frequency and document frequency components in the proposed approaches, the mixed-query baselines assigned weights of technical terms independently from weights of their translation(s), although these translations were monolingually structured, thus resulting in the deterioration of performance. Furthermore, the over-weighting problem makes both the lower baseline and the search-engine-retrieval-like runs yield significantly worse results,

compared to proposed re-weighting schemes, and makes the mixed documents, which are relatively poor compared to English documents, dominating the top ranked list.

Compared to upper baseline, the effectiveness of the two proposed re-weighting approaches (CRSQM-NODECAY and CRSQM-DECAY), illustrated in Figure 6.4, showed good results for the CRSQM-DECAY run, which achieved more than 94% of monolingual baseline effectiveness, and approximately 88% for the CRSQM-NODECAY run.

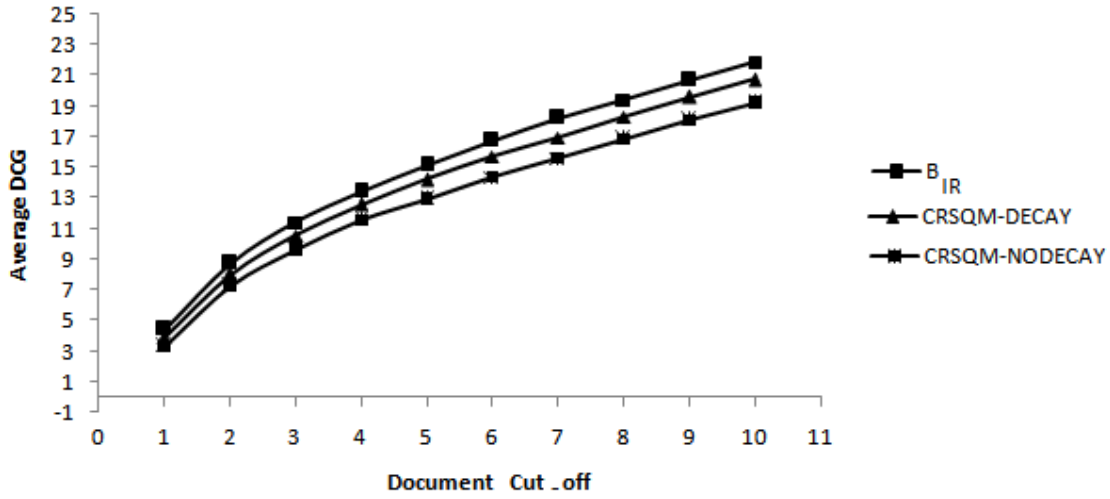


FIG. 6.4: The diagram compares retrieval effectiveness, in terms of average DCGs, of the monolingual upper baseline (b_{IR}) and the proposed cross-lingual structured query model, with and without using a damping weight factor for co-occurring bilingual terms, which are the (CRSQM-DECAY) and the (CRSQM-NODECAY) runs, respectively.

Extending the term frequency statistics of the cross-lingual structuring and re-weighting to consider the phenomenon of any two similar and bilingual terms that co-occurred together in documents (i.e. 'deadlock الإقفال', in which the term 'deadlock' co-occurs with its Arabic translation), suggests that such neighbouring terms in different languages, even within a predefined window, can have a substantial effect on retrieval performance. This is obvious when the average DCG values of the CRSQM-NODECAY weighting are compared with those in the CRSQM-DECAY weighting. The improvement in retrieval performance between the runs at top 10 was distinguishable and statistically significant (p-value < 0.000012). Indeed, the CRSQM-DECAY run also outperforms the two mixed-query baselines. This moderate improvement in CRSQM-DECAY derived from the fact that both the prior well-established cross-lingual weighting in CRSQM-NODECAY (and the lower baseline as well) may cause some terms, even when they are cross-lingually structured, in the mixed merged queries to earn somewhat double weights, due to the co-occurrence of the same term in multiple languages in document. In the CRSQM-DECAY run the cross-lingual term frequency suppresses the impact of such co-occurred pairs into different languages.

However, the difference in performance in both the CRSQM-DECAY and the CRSQM-NODECAY runs was not consistent through all queries. Figure 6.5 illustrates a query-by-query comparison for some of the queries using the CRSQM-NODECAY approach versus the CROSS-DECAY method. Queries are numbered

for the presentation. The majority of the queries (particularly 15 out the shown 18 queries) in the figure show better results or sustain the same performance, for CRSQM-DECAY over the CRSQM-NODECAY run.

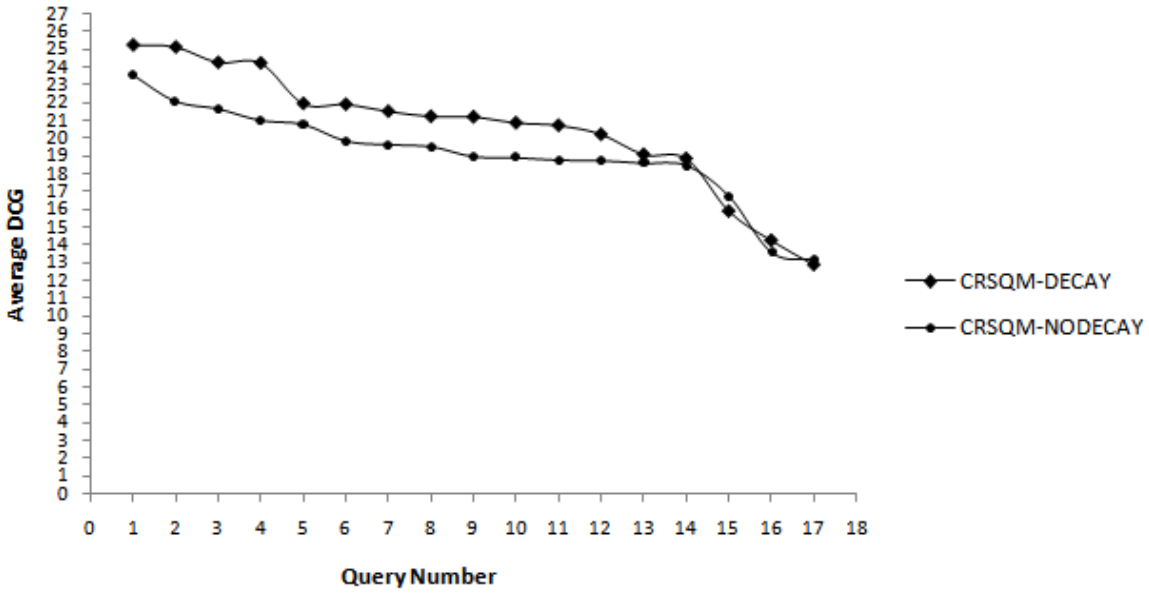


FIG. 6.5: Query-by-query comparisons, in terms of average DCGs, of some topics in MUMIXEAC collection for the proposed CRSQM model, with and without using a damping weight factor for co-occurring bilingual terms, which are the (CRSQM-DECAY) and the (CRSQM-NODECAY) runs, respectively.

This was primarily derived from the artifact that a considerable number of technical terms in many mixed documents contain different snippets/phrases/terms but into two different languages. Note that the tuning parameter value of k_1 in the used okapi BM25 model in all experiments was set to the value 2. In Figure 6.5 it is shown also that the performance, in terms of an average DCG, was indistinguishable and almost similar for some queries, especially for some of the first queries, but it was significant for others. There are only few cases in which the CRSQM -NODECAY run outperformed the CRSQM -DECAY (3 queries in the plotted graph). With respect to these queries, this was mainly because some English keywords, e.g., the word 'system', can be found in many documents but in different topics. Thus, if the term frequency of such a word is high in some irrelevant documents, this may probably result in the inclusion of these documents on the retrieval list, possibly at the top, but depending on their frequencies. The findings of the proposed re-weighting schemes imply major observation. Firstly, there are a considerable number of mixed documents that include terms that were written into two different languages. Handling these terms reasonably can result in a significant effect on performance.

6.2.1.3 Study III: Weighted Inverse Document Frequency

While the type of over-weighting that was dealt with in the previous study is mainly caused by mixture of languages in mixed documents and individual computations (both term frequency and document frequency) of terms that are cross-lingually similar, traditional overweighting is merely caused by incomparable sizes of sub-collections. More specifically, the former type of overweighting resulted from the biased document frequency (see section 1.1.2.1) of similar cross-lingual terms. Instead, traditional overweighting can occur even if document collection consisting of several monolingual documents/sub-collections only.

Using a damping factor for index terms in each sub-collection may minimize traditional overweighting and moderate that one which occurs due to mixture of texts. This damping factor is based on estimating a relative weight for each sub-collection in terms of its size with respect to the size of the entire collection, which is both mixed and multilingual. Thus, this study, which was called the WT-IDF (stands for WeighTed Inverse Document Frequency), was focused on the impact of using such a damping factor for a sub-collection, with regards to mixed-language querying, when it is incorporated with some well-known techniques such as conventional structured query models.

Aims

With respect to mixed-language querying, overweighting is big drawback. Therefore, study III dealt with over-weighting (if it is monolingual or caused by mixture of texts), which rose when a single index for all documents, regardless of their languages, is used. It seeks to evaluate if the combination between the proposed weighted inverse document frequency, in terms of a damping factor for weights of terms derived from their corresponding sub-collections, and the traditional approaches of structured query model can have a significant effect on retrieval performance of mixed-language queries. Furthermore, to what extent is the assumption valid that a sub-collection with a higher number of documents is more significant than another sub-collection with a small number of documents.

Methodology

To mitigate the potential problematic weighting due to indexing all documents in different languages into a single centralized pool, a damping weight factor for terms is computed from sub-collections weights. This damping factor should be considered as just an additional parameter to measure the usefulness of each sub-collection in which the term occurs, with the assumption that a sub-collection with a higher number of documents is more significant than another sub-collection with a smaller number of documents. Accordingly, during indexing time of documents, the number of documents in each monolingual/mixed sub-collection is determined and the weight is computed for each sub-collection as in equation 4.7.

Thus, having established a mixed merged query, as was described in the common methodology of the section, all the alternative translations, mostly in Arabic, for a certain source query term whose translation(s) was found in the technical dictionary, are structured, but using the traditional monolingual structured query of Kwok approximation (equation 2.30 for term frequency and equation 2.33 for document frequency). This would result in creating instances of the translations of the source query term in a single language. The cross-lingual structuring approach was not utilized in this study.

Following this, for each term (including those monolingually structured, which are handled as instances of a single term) in the mixed merged query, the damping factor of each sub-collection is then incorporated in the document frequency computations of that term as in equation 4.8. This strategy was applied to all terms. It was not limited to technical terms only or terms whose translation were obtained from the technical dictionary. Instead, it was applied to each term in the mixed merged query, regardless of any prior criterion, as the major purpose here was to avoid overweighting (traditional or caused by mixture of texts). For instance, for the Arabic terms the weight of the Arabic sub-collection was retrieved and then it was merged in their DFs. This modification of term weights in terms of damping weight factor is repeated for each term in the big mixed and merged query, documents scores were computed and the final list was retrieved. However, it should be noted that the second method, which is the relative frequency factor in section 4.2.3.2, can be also used as it has the same effect on final inverse document frequency.

Results and Discussion

Figure 6.6 on the top of the next page shows the results of the WT-IDF in terms of average DCG curve, compared to the b_{CLIR} baseline and the best results that had been obtained thus far, which was for the CRSQM-DECAY run. Table 6.4 on the next page also, lists the corresponding results in tabular form.

In the figure, it is obvious that the combination of the damping weight factor of each sub-collection with the monolingual structured query model in establishing a weighted IDF fared well, outperforming the CLIR lower baseline (b_{CLIR}). In particular, the WT-IDF run yielded statistically significant better results over the lower base (p-value =0.029028), based on a paired two-tailed Student's t-test with $p<0.05$.

Measure	Average DCG @									
Run	1	2	3	4	5	6	7	8	9	10
WT-IDF	3.532	6.447	8.648	10.35	11.697	13.097	14.37	15.427	16.474	17.338
CRSQM-DECAY	3.915	8.02	10.551	12.540	14.217	15.731	16.959	18.292	19.561	20.746
b_{CLIR}	3.340	6.277	8.612	10.357	11.640	12.759	13.904	14.783	15.770	16.641

TABLE 6.4: Effectiveness evaluation, in terms of average DCGs, of the weight inverse document frequency run, the lower baseline (b_{CLIR}) and the cross-lingual structured model (CRSQM-DECAY).

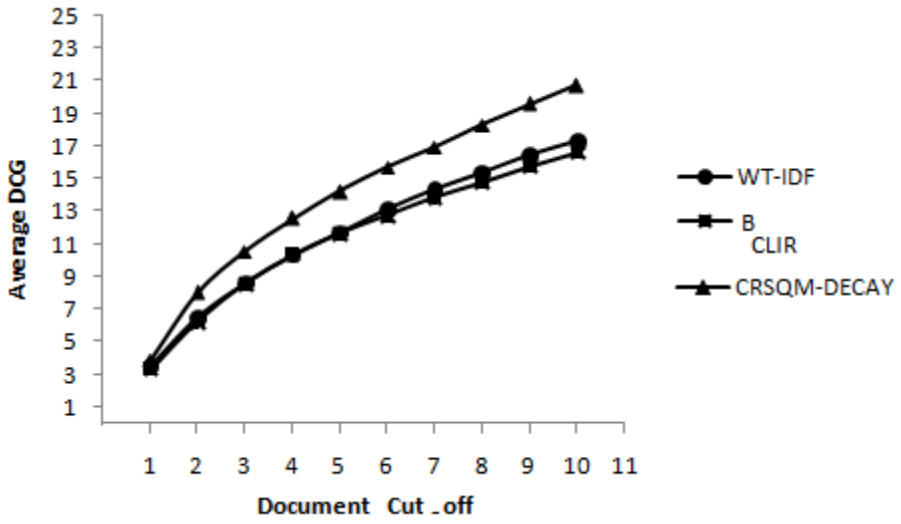


FIG. 6.6: Retrieval effectiveness, in terms of average DCGs, of the weighted inverse document frequency run (WT-IDF), the cross-lingual lower baseline (b_{CLIR}) and the proposed cross-lingual structured query model, with a damping weight factor for co-occurring bilingual terms (CRSQM-DECAY).

The reason for the improvement in the effectiveness is that the individual re-weighted IDF factor of each term, regardless of its language, was utilized to suppress its importance and moderate its overweighting. Since, the boosted values in the over-weighted terms hinged on the total number of documents in each corresponding sub-collection, the WT-IDF would probably result in lower diminished values for the English terms while the diminishing values for the Arabic terms would be high. In particular, weights of Arabic terms, which cause the significant overweighting, would have smaller IDF values (and thus lower weights) than English terms, when the weighted IDF approach is used. The IDF factor was thereby used to make a difference in weights and consequently makes the WT-IDF run outperforms the b_{CLIR} baseline. The situation is somewhat similar to the trend of the previous study (study II). However, since the top documents in both the two runs (b_{CLIR} and WT-IDF) were mainly obtaining their weights depending on TF and due to the accumulation of values in the used DCG measure, the differences in these top documents begin with small values and the effect of the weighted IDF in the WT-IDF run does not reflect immediately. It is also important to note that the contribution of the over-weighted Arabic translation of technical terms to final estimated scores of documents in both the WT-IDF and the lower baseline was firstly suppressed by applying the Kwok approximation monolingually (Arabic translations of technical terms were structured and clustered together). This would likely reduce the magnification of the over-weighting in conventional centralized indexing, in which the IDF is computed individually for each Arabic translation, instead of structuring all translation alternatives together – as in both the weighted inverse document frequency and lower baseline runs. Nevertheless, such contribution in lowering the impact of overweighting was not the major reason for the effectiveness improvement of the WT-IDF, as the same approach of monolingual structured was also utilized for the lower baseline run, too. The improvement in performance was merely caused by the use of the damping factor of sub-collections.

Comparing improvement performance of the WT-IDF with the best obtained result in the previous study (CRSQM-DECAY) in Figure 6.6, it is obvious that the latter effectiveness improvement was unreachable by the former re-weighting approach. In fact, the difference between the two approaches is statistically significant (p-value < 0.000003) for the CRSQM-DECAY. Examining the search results exposed two observations and provides some additional insight into how both the CRSQM-DECAY and the WT-IDF impact can differ. On one hand, although the impact of the over-weighted Arabic terms was moderated in the WT-IDF run using both monolingual structuring and the weighted IDF for mixed query terms, these terms were still over-weighted. Since mixed documents cannot be attributed to a certain language (any query term in the mixed query, Arabic or English, can fairly occur in a monolingual or a mixed document), Arabic terms benefit from the number of mixed documents in the collection, which was relatively high compared to this monolingual Arabic sub-collection (number of documents in mixed sub-collection size is approximately 36 times the number of monolingual Arabic documents). This is not the case for the English terms, although they were boosted too, which were expected to have high document frequency as the number of monolingual English documents was high (number of documents in English sub-collection size is approximately 3 times number of mixed documents). Thus, Arabic terms would be having higher weights which results in minimizing the number of English monolingual and highly relevant documents at top ranks. This fact, besides the independent computation of weights cross-lingually as well as the ignorance of co-occurrence of neighbouring terms in different languages, contributes to minimizing the WT-IDF effectiveness.

6.2.1.4 Study IV: Hybridized Cross-lingual SQM with Weighted IDF

The two prior studies reported (study II and study III) were conducted individually. The empirical result of each in isolation was promising. Thus, study IV, which was called CRSQM-DECAY-WT-IDF, tested the impact of using a hybrid approach of the two algorithms.

Aims

This study aimed to test the impact of combining the two techniques described above (cross-lingual structured model and weighted IDF) and whether this combination can significantly improve retrieval of mixed-language querying. Furthermore, as cross-lingual structuring of queries can result in high DF, and thus low weights, the study aimed to show whether the effect of the over-weighted *non-technical* terms regardless of their languages, in mixed merged queries is reduced whenever the cross-lingual structured model is utilized.

Methodology

The combination of the two approaches (cross-lingual structured model and weighted IDF) was been implemented sequentially. The methodology for the CRSQM-DECAY run, as shown in section 6.2.1.2 in

study II, resulted in a cross-lingual structure that grouped all the alternative translations for a certain technical source query term and the source term itself. For source terms that tend to co-occur with their equivalents in another target language, a decaying factor was used, as in the methodology of the CRSQM-DECAY run.

Next, a damping weight factor was computed for each monolingual sub-collection, as shown in the methodology section of study III. Following this, for each term in the big and mixed merged query, including those being cross-lingually structured (but handled also as a single cross-lingual term), the damping weight of each sub-collection was then incorporated in the document frequency computations of that term, as illustrated in the previous section.

Again, this strategy was applied to all terms and it was not limited to technical terms only or terms whose translations were obtained from the technical dictionary. The combined approach was applied to all terms in the mixed merged query. Documents scores were then computed and a final list was retrieved as a result.

Results and Discussion

Table 6.5 on the next page provides the findings of the hybrid system of combining the CRSQM-DECAY and the WT-IDF approaches, compared to the composing approaches themselves and both the lower and the upper baselines. The diagram in Figure 6.7 plots the same data in terms of average DCG score curves. The effect of appropriately combining the two approaches on retrieval, compared to WT-IDF, is statistically significant. The substantial increase in average DCG was 3.998 at the rank position 10, with a p-value < 0.000002. Clearly, there is some agreement with the results in the CRSQM-DECAY run, which showed an observable increase in retrieval performance when cross-lingual structuring of queries was introduced. This means that the major impact on performance was derived merely from the use of the CRSQM-DECAY, rather than from the WT-IDF. This leads to an interesting observation concerning the role of the weighted IDF in the CRSQM-DECAY-WT-IDF versus the WT-IDF run, which is also shown in Figure 6.7.

Using a weighted IDF strategy in CRSQM-DECAY-WT-IDF run did not have any impact on the technical terms. This is due to the cross-lingual structuring of these technical terms. In particular, for terms that were considered as synonyms across languages (those were cross-lingually structured and almost technical), their modified weights using the CRSQM-DECAY would likely be kept, when the weighted IDF strategy starts its work. This is because a term that is cross-lingually structured cannot be attributed to any sub-collection (thus, no damping weight factor would be applied) and, hence, their IDFs were computed across the entire multilingual collection, resulting in weighted IDFs that won't have any effect on these cross-lingual, and yet technical, structured terms. Consider a cross-lingual set containing the terms deadlock and الإقفال. Both of the terms or one of them can occur in a monolingual Arabic document, a monolingual English document or a mixed document. Thus, the damping factor for such cross-lingual terms would likely be 1, as it was discussed in section 4.2.3 in the design chapter.

Measure	Average DCG @									
Run	1	2	3	4	5	6	7	8	9	10
CRSQM-DECAY-WT-IDF	4.255	8.468	11.086	13.096	14.865	16.396	17.578	18.855	20.177	21.336
CRSQM-DECAY	3.915	8.020	10.551	12.540	14.217	15.731	16.959	18.292	19.561	20.746
WT-IDF	3.532	6.447	8.648	10.350	11.697	13.097	14.370	15.427	16.474	17.388
b_{IR}	4.447	8.745	11.403	13.424	15.174	16.771	18.211	19.409	20.671	21.882
b_{CLIR}	3.340	6.277	8.612	10.357	11.640	12.759	13.904	14.783	15.770	16.641

TABLE 6.5: Results of the proposed combination of the cross-lingual structured query model and the weighted inverse document frequency approach (CRSQM-WT-IDF), compared to its composing approaches (CRSQM-DECAY and WT-IDF). Values are presented in terms of average DCG and they are compared also to those obtained by the upper baseline (b_{IR}) and the lower baselines (b_{CLIR}).

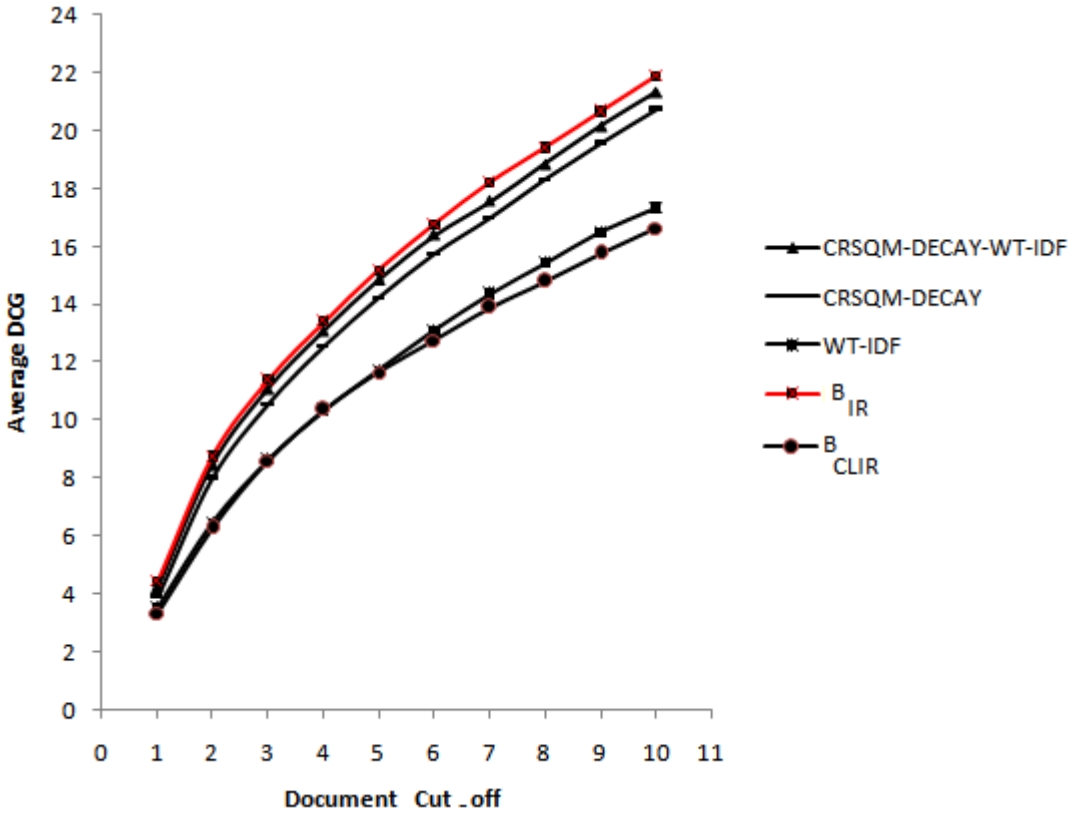


FIG. 6.7: Retrieval effectiveness, in terms of average DCGs, of the combined approach of the cross-lingual structured model with weighted IDF (CRSQM-DECAY-WT-IDF) and its constituent approaches (CRSQM-DECAY and WT-IDF). Curves are also compared to those of the cross-lingual lower baseline (b_{CLIR}) and the monolingual upper baseline (b_{IR}).

Bearing in mind this observation, the utilization of the weighted IDF in the CRSQM-DECAY-WT-IDF run was to minimize the impact of non-technical terms, rather than technical, which may cause the system to skew, even slightly, the list of the retrieved documents towards these non-technical terms. This is especially true as the cross-lingual structuring of technical terms often results in a high joint DF (low weights) and the non-technical terms were already over-weighted, whether they are in English or Arabic. However, since the cross-lingual structuring was applied first, the impact of the weighted IDF on non-technical terms would likely drop to a low level. This is especially true if the most significant parts in mixed merged queries were technical and those that are non-technical were reduced with the use of the stop lists in the two languages. This resulted in lower contribution for the weighted IDF to the improvement obtained by the CRSQM-DECAY-WT-IDF.

Different to this behavior, the effect of the weight inverse document frequency in the WT-IDF run alone is to reduce the impact of the over-weighted terms , regardless of their languages and regardless of whether they are technical or not. Thus, the role in the WT-IDF is to keep all terms closer to their original IDF values as if sub-collections were not merged together into a single multilingual collection. The findings of the CRSQM-DECAY-WT-IDF weighting in Figure 6.7, compared to its partially constituent CRSQM-DECAY, did show minor improvement on retrieval effectiveness (CRSQM-DECAY-WT-IDF achieved better score at top 10 documents, but the difference in the average DCG was 0.59034). The p-

value is statistically significant (p-value = 0.012859), but it is not high. This is because the cross-lingual structuring minimizes the effect of the overweighting in technical terms, which are the major keys in searching, as was argued above.

Compared to the upper baseline, the performance of the CRSQM-DECAY-WT-IDF reaches a creditable 97.507% of the monolingual baseline effectiveness and records statistically significant improvements over the mixed query-based baselines. These empirical results confirms the fact that the monolingual assumption of weighting and retrieval in mixed queries may lead to inconsistent weights, thereby creating greater scope for skewing weights in mixed documents.

6.2.2 Experiments Using Mixed-Languages in separate Indices

The following section describes the experiments that were conducted to evaluate some of the developed techniques, particularly those in section 4.3, to handle mixed-language queries and documents in a separate indices environment. Specifically, for indexing documents in this part, experiments utilized a combined architecture that combines the two basic architectures of indexing (distributed and centralized) while considering the mixed document characteristics and their weightings. Particularly, as this combined architecture makes use of a centralized index, in addition to the distributed architecture, an overweighting problem may occur due to the mixture of texts in documents (it causes a biased document frequency problem), a re-weighting component, which was placed in the centralized index of the proposed combined architecture of indexing, was utilized.

Since documents were not placed together into a single index in the proposed combination of indexing, then it is possible to consider the entire approach as a traditional distributed architecture, despite the insertion of the centralized index. Due to this distributive feature in indexing, documents in the proposed indexing architecture, some merging methods were used to produce the final ranked list.

Accordingly, this section consists of four experiments that were grouped and discussed together in a one study, namely study V. All of the four experiments made use of the proposed indexing and weighting in section 4.3, but they were different in their merging methods.

6.2.2.1 Study V: Combined Architecture with Cross-lingual SQ Model

Aims

The main objective of study V is to test if the major drawbacks of both of the two conventional approaches of indexing with regards to both mixed queries and mixed and multilingual collections can be suppressed, whenever the proposed hybrid indexing approach is used in a combination with the proposed probabilistic cross-lingual structured query model.

Traditional distributed approaches are usually compared to centralized architecture. This approach has been substantially used, for example, Rasolofo et. al (2001) and Lin and Chen (2003). Therefore, one of the major research question for this study is what is an efficient indexing approach for managing the

mixed-language problem (mixed-language IR system)? Is it the centralized architecture, which is used as a baseline, or the distributed architecture?

Although, exploring merging algorithms is not the focus of this thesis, they have a significant impact on effectiveness. Therefore, this study also aimed to compare briefly the effectiveness of the proposed solutions, which are the hybrid indexing approach and the probabilistic cross-lingual SQ model, when different merging methods were utilized.

Methodology

Since the architecture was basically distributed, three different indices of two types were created. The first type consists of two monolingual and distributed indices - one was in Arabic and the second was in English and the second type contains a single mixed and centralized index (see Figure 4.2). In particular, in the first type of monolingual indices, monolingual English documents were placed into a separate monolingual index, while monolingual Arabic documents were put into another separate monolingual index.

In each of the two monolingual indices, terms were extracted, normalized and stemmed according to their corresponding languages, as illustrated in sections 6.1.1 and 6.1.2. Two fields only, which were the TITLE-Arabic and the CONTENTS-Arabic, among the four previously identified ones were used in the Arabic index to populate the Arabic terms, whereas in the English index the TITLE-English and the CONTENTS-English fields were employed. This resulted in two distributed monolingual indices.

It was previously shown in section 6.1.3 that two monolingual queries were often produced after translating each original source mixed query. Thus, during retrieval of a certain query, each of the two indices was searched directly with its corresponding monolingual queries, e.g., a monolingual English index was searched by a monolingual English query. Thus, two intermediate, individual and monolinguals lists, one in Arabic and the second in English, were obtained for each corresponding submitted query.

Mixed documents were placed into a centralized mixed index. Since the architecture used here was basically centralized, both the translated monolingual queries, which were obtained from translation process were merged together to form a big and mixed merged query, as in the previous studies.

Next, the proposed probabilistic cross-lingual structured query model in section 4.3.3 was used to estimate term frequency component (equation 4.19), document frequency (as in equation 4.6, which is the proposed cross-lingual model of the Kwok approximation) and document length (equation 4.21). The impact of applying these equations is a cross-lingual structuring of terms, whose translations were obtained from the technical dictionary, in the big and mixed merged query.

Eventually this merged query was submitted to the centralized and mixed index and another third individual, but mixed, list was obtained.

The three individual lists of each single query in its different forms, which were produced from the three indices of the proposed combined index, were merged together as previously illustrated in Figure 4.3.

The first employed merging method was the raw score (see section 2.3.2 in the review chapter of CLIR). This run was called COMB-PCSQ-RAW.

In the second experiment, which was called COMB-PCSQ-MAX, scores in each individual list were firstly normalized by the maximum score in that list and then results of all lists were merged together according to the new normalized scores.

The third experiment, which was called as COMB-PCSQ-MINMAX, followed the same methodology but both the minimum and the maximum scores in each individual list were used to normalize scores, as in equation 2.38.

In the fourth experiment, which was called COMB-PCSQ-CORI, scores in each list were normalized by the CORI approach of Rasolofo, et al. (2001), in which the premise is that if a certain sub-collection contains more retrieved documents, then this sub-collection probably contains more relevant documents (see section 2.3.2).

Accordingly, each sub-collection's score is computed with respect to the proportion of documents retrieved (the length of the result). That is:

$$Sc_k = log\left(1 + \frac{l_k * C}{\sum_{i=1}^{|M|} l_i}\right) \tag{6.1}$$

Where Sc_k is the score of the kth sub-collection, M is the total number of all sub-collections, l_k is the total number of documents that are retrieved by this kth sub-collection and C is a constant for normalizing the score of the sub-collection. Using this sub-collection score, its weight is computed as follows:

$$wt_k = 1 + \left[\frac{Sc_k - avg\_Sc}{avg\_Sc}\right] \tag{6.2}$$

Where Sc_k is the above computed score of the kth sub-collection and avg_Sc is the mean sub-collection score. Finally, the scores of documents in each sub-collection are computed by multiplying the weight of each sub-collection by the original scores of documents in that sub-collection as illustrated before in the CORI approach, specifically scores are computed according to equation 2.39.

Results were then ranked according to the produced scores. The four experiments are referred to as *combined-index-based methods*. After a single unified list for each query was produced, the top 10 documents were used to measure performance using the discount cumulative gain and results were also averaged, as in the previous studies.

Results and Discussion

Values shown in Table 6.6 are the average DCG at top 10 documents across the 47 queries for the four runs described in the methodology section (COMB-PCSQ-RAW, COMB-PCSQ-MAX, COMB-PCSQ-MINMAX and COMB-PCSQ-CORI). All findings were compared on the same table to the lower baseline (b_{CLIR}) generated in the centralized index in study I.

Measure	Average DCG @									
Run	1	2	3	4	5	6	7	8	9	10
COMB-PCSQ-RAW	3.240	6.88	8.924	10.764	12.263	13.547	14.958	16.145	17.356	18.416
COMB-PCSQ-MAX	3.970	6.35	7.515	9.346	10.827	12.684	13.732	14.688	15.676	16.543
COMB-PCSQ-MINMAX	3.970	6.35	7.515	9.816	11.873	12.964	14.089	15.296	16.344	17.106
COMB-PCSQ-CORI	3.340	6.96	9.181	10.821	12.457	13.804	15.257	16.71	17.934	19.066
b_{CLIR}	3.340	6.277	8.612	10.357	11.640	12.759	13.904	14.783	15.770	16.641

TABLE 6.6: Results of the proposed hybrid architecture for indexing engaged with the probabilistic cross-lingual structured query model and one of the merging methods in traditional distributed architecture. These are the (COMB-PCSQ-RAW, COMB-PCSQ-MAX, COMB-PCSQ-MINMAX and COMB-PCSQ-CORI) runs. Results are compared to the baseline run (b_{CLIR}). Values are presented in terms of average DCG computed at document cut-off values [1..10] for 47 queries in the MULMIXEAC test collection.

The same results are also shown, for presentation simplicity, in Figure 6.8 and Figure 6.9. In particular, the diagram in Figure 6.8 depicts the results of retrieval performance of the COMB-PCSQ-RAW and the COMB-PCSQ-CORI runs, compared to the lower baseline in a graph, whereas Figure 6.9 shows

performance effectiveness of the COMB-PCSQ-MAX and the COMB-PCSQ-MINMAX runs, compared to the same baseline. Figure 6.10 shows the results of four methods compared to lower baseline.

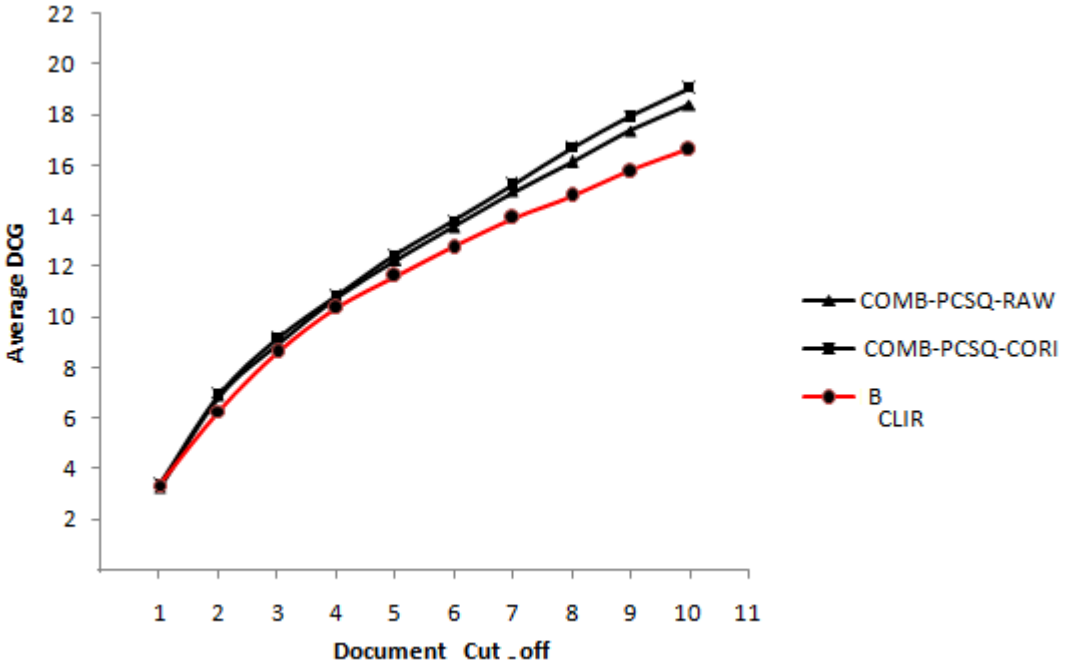


FIG. 6.8: Retrieval performance of the proposed combined architecture with the probabilistic cross-lingual structured query model engaged with the raw and CORI merging methods, COMB-PCSQ-RAW and COMB-PCSQ-CORI, respectively. Results are compared to those obtained by the lower baseline (b_{CLIR}).

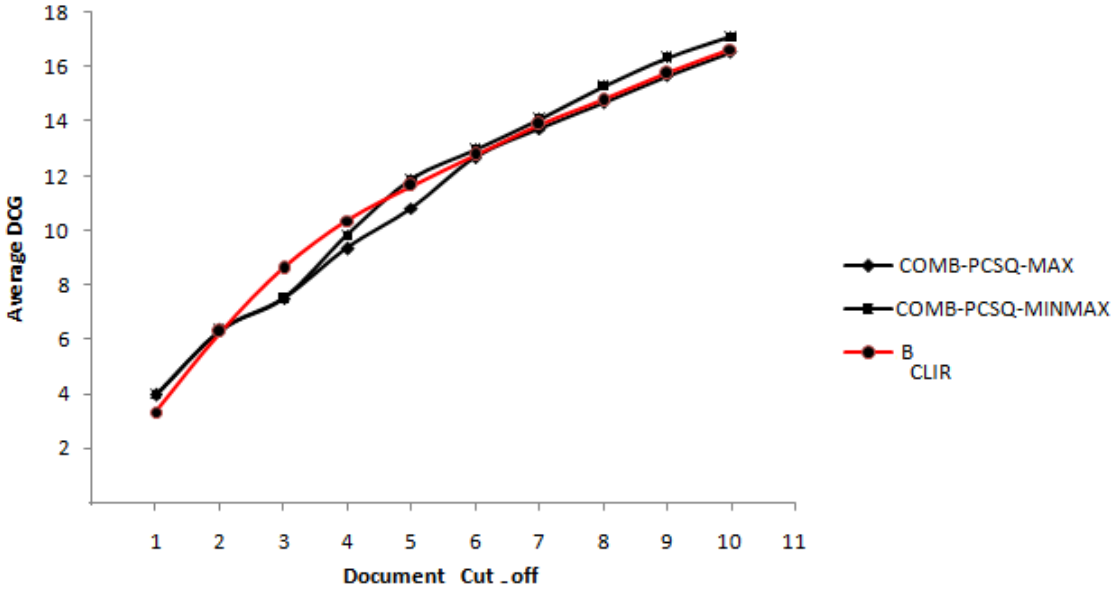


FIG. 6.9: Retrieval performance of the proposed combined architecture with the probabilistic cross-lingual structured query model engaged with the merging methods which normalize scores through maximum scores adjustment (COMB-PCSQ-MAX) and both maximum and minimum scores (COMB-PCSQ-MINMAX) adjustment. Results are compared to those obtained by the lower baseline (b_{CLIR}).

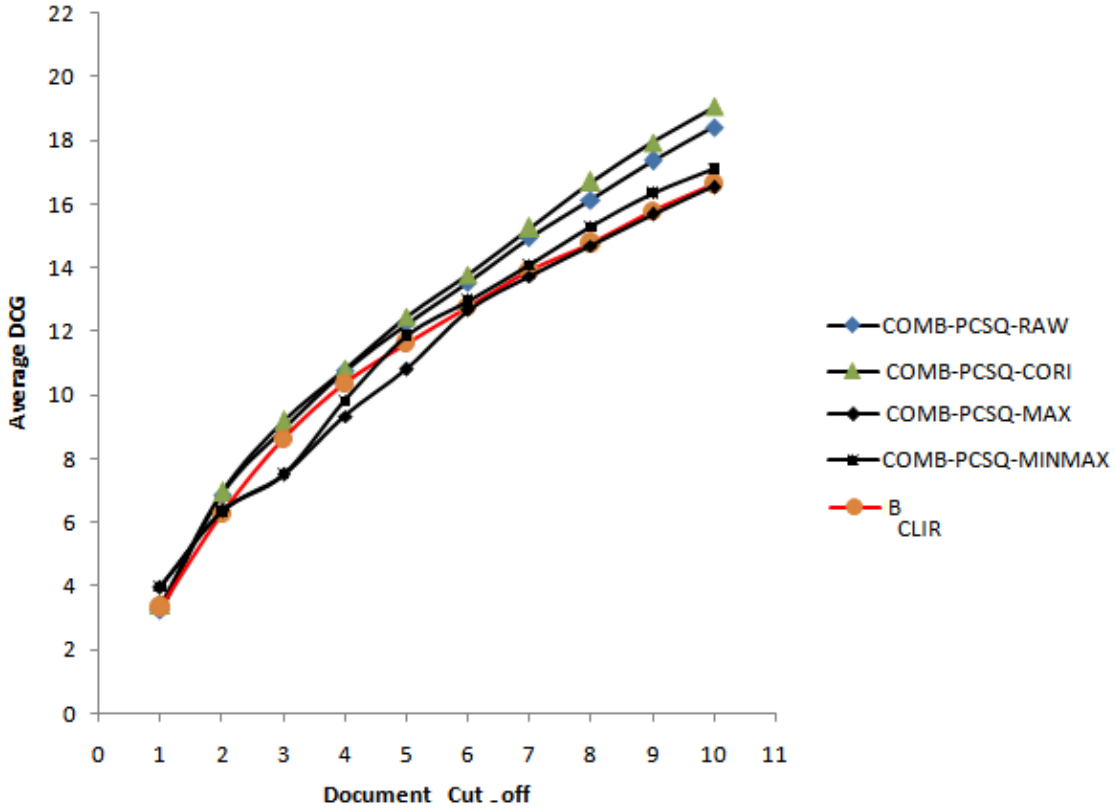


FIG. 6.10: Retrieval performance of the proposed combined architecture with the probabilistic cross-lingual structured query model engaged with the raw (COMB-PCSQ-RAW), CORI (COMB-PCSQ-RAW) merging methods, the merging methods which normalize scores through maximum scores adjustment (COMB-PCSQ-MAX) and both maximum and minimum scores (COMB-PCSQ-MINMAX) adjustment. Results are compared to those obtained by the lower baseline (b_{CLIR}).

Figure 6.8 shows clearly that coupling of the combined architecture of indexing and weighting with both of the utilized raw and CORI merging methods outperforms the basic lower baseline in its performance effectiveness, with the best retrieval related to the COMB-PCSQ-CORI. The p-values of these two combined-index-based methods, compared to lower baseline were statistically significant (the two p-values were 0.00129 for COMB-PCSQ-CORI and 0.000158 for COMB-PCSQ-RAW). This difference in performances was due to the reasons discussed next.

First of all, the use of the distributive feature as a major approach for indexing in the proposed combined architecture caused the combined-index-based methods, in general, to minimize the monolingual over-weighting problem because the document frequency component for terms will not increase. This was not the case in the lower base, which suffers from this problem as documents were placed into a single index. Furthermore, the adoption of the centralized approach in the combined architecture for indexing only Arabic-English mixed documents causes the proposed architecture to artificially avoid partitioning these mixed documents across the two monolingual sub-collections/languages and thus, the score of every relevant mixed document was reasonably weighted (as the over-weighting was minimized and the cross-lingual structuring was used) and fairly computed (because content was not divided).

The same phenomenon of fair competition of documents appeared also with the adoption of two monolingual indices in the combined architecture. Since mixed documents are not placed in these monolingual distributed indices, the overweighting problem did occur. This makes the performance of these monolingual runs better. Note that the used monolingual queries in these distributed monolingual indices were partially, instead of completely, translated as the original source queries were mixed and, thus, translation disambiguity was reduced to lower levels.

It was previously shown that if a mixed document is partitioned using a traditional distributed architecture, then it will not compete, even if it is highly relevant, in the sub-collections in which this mixed document is partitioned (see section 4.3.1). It was also described that monolingual documents will not compete also if a traditional centralized index is used for all documents – as only a portion of the query would match its content, unlike mixed documents whose scores would be computed from the entire mixed and merged query. Accordingly, placing mixed documents in a separate centralized pool, as in the combined indexing approach, opens the road for the highly relevant (monolingual or mixed) documents to compete. This is because after each sub-index in the hybrid indexing approach returns the intermediate result, all these results will be merged together and, thus, each intermediate result will participate with some documents, at least from its top documents, in the final unified list.

Furthermore, the use of the probabilistic cross-lingual structured query model for re-weighting terms in the inserted centralized index of the combined architecture, minimizes the over-weighting that occur due to mixed-language feature in documents. Thus, cross-lingual structuring contributed to the better performance of the CORI and raw combined-index-based methods, specifically, over the lower baseline run and contributed in the performance also in the combined-index-based methods, in general.

Examining the intermediate results in each sub-collection/index in the combined architecture (distributed Arabic, distributed English and centralized mixed in both Arabic and English) revealed that the performance of the English run was much better than the mixed intermediate run, which in turn was better than the Arabic retrieval. This showed clearly that English documents are more valuable than both mixed and monolingual Arabic documents.

The results in both Table 6.6 and Figures 6.8, 6.9 and 6.10 showed also that the performance of the developed combined architecture and cross-lingual structuring re-weighting with different merging methods was varied. In particular, the performance of the proposed solutions with merging approaches that normalized scores with the maximum score in each list (COMB-PCSQ-MAX) and with both minimum and maximum scores (COMB-PCSQ-MINMAX) were worse than the performance of the same proposed solutions with the raw score and the CORI merging methods. In fact, the latter methods have had a similar performance to the lower baseline, with a slight difference in the average DCG (particularly 0.465) for the COMB-PCSQ-MINMAX run over baseline at rank 10 and a difference of 0.098 for the baseline in the case of the COMB-PCSQ-MAX run.

Although the focus of experiments here was on the proposed solutions, rather than the merging approaches, this phenomenon has some primary and secondary reasons. The secondary reason was the bad retrieval of the Arabic monolingual sub-collection. Particularly, examination of individual lists showed that the performance of the monolingual Arabic retrieval was much worse and the quality of

monolingual Arabic documents was usually poor. One possible reason for this bad retrieval, beside the small size of the Arabic sub-collection, is the synonymy feature of the Arabic language, in which a single word may have several probable meanings, e.g. الإقفال, which means deadlock in computer-based vocabulary, and thus, it cannot easily disambiguate unless the context is used. Accordingly, the monolingual Arabic retrieval reduced effectiveness markedly and dropped retrieval performance for both COMB-PCSQ-MINMAX and COMB-PCSQ-MAX runs in the combined-based approaches, when intermediate ranked lists were merged together.

The primary reason for the relatively worse performance of COMB-PCSQ-MINMAX and COMB-PCSQ-MAX runs, compared to COMB-PCSQ-RAW and COMB-PCSQ-CORI, was the strategies that are often adopted by these two merging methods. For example, if the maximum score in a single ranked list is much higher than the maximum score in a second list, both the top scores will be normalized to 1 or close to this value, when scores are normalized using both the minimum and the maximum scores or the maximum score only in each individual list. Thus, each individual list will have at least some of its top documents in the final ranked list after merging. But, with the bad retrieval of the monolingual Arabic sub-collection, this would likely result in favouring some documents in this list, although the latter (Arabic retrieved list) have some documents with lower scores on the top ranks. For instance, in the many queries of the monolingual Arabic sub-collection, intermediate results in both the COMB-PCSQ-MINMAX and the COMB-PCSQ-MAX runs participate with two documents on average (more than 20% of the top ranked documents) in the final ranked list, when their individual intermediate results were merged with the other retrieved lists. A similar trend was also shown by Lin and Chen (2003), who illustrated that if the score of the top document (maximum score) in a list is much greater than the one that follows on the same list (second document), then the normalized score of latter document (second document) would be low even if its original score is high. Thus, the final rank of this document would be lower than that of the top ranked documents with very low but similar original scores in another result list.

For these reasons, the accuracy performance of both the COMB-PCSQ-MINMAX and the COMB-PCSQ-MAX runs in Figures 6.9 and 6.10 was the best at top document (see also values in Table 6.6), compared to the accuracy of COMB-PCSQ-RAW and COMB-PCSQ-CORI. In particular, as the former methods normalized the first document in each individual rank list to the value 1 and the merging process to produce the final ranked list started from the English retrieved documents, which were monolingual searched by a partially translated monolingual query, in which the significant terms were originally written by the user in English (those are technical in the original mixed query). Thus, the first document is likely predicted to be highly relevant. This was not the same merging mechanism in the COMB-PCSQ-RAW, for example, which was probably unfair because of using the raw scores of the retrieved documents of the mixed centralized index, which was being searched by long and mixed merged queries, preferring these documents over monolingual ones.

The same reason of normalizing the top document of each individual list to the value 1 in the COMB-PCSQ-MINMAX and the COMB-PCSQ-MAX runs caused also a significant deterioration of retrieval, especially at rank 3 in Figure 6.9, which was exactly the documents of the Arabic retrieval, as the

merging process was performed by the English list firstly, mixed list secondly and the Arabic list eventually, whenever values are equaled. After the first half of the merged documents, approximately, the retrieval performance in both COMB-PCSQ-MINMAX and COMB-PCSQ-MAX runs became relatively better. This is because there were some queries that didn't retrieve any document in the monolingual Arabic retrieval or only very few documents (3 in some cases) were returned. Due to these drawbacks, both the two runs (COMB-PCSQ-MINMAX and COMB-PCSQ-MAX) had lower retrieval performance, when they were compared to COMB-PCSQ-RAW and COMB-PCSQ-CORI runs. Thus, the degradation in performance stemmed from the merging methods, rather than the proposed solutions (combined architecture with PCSQ).

The best retrieval results in the combined-index based methods were yielded by the proposed solutions with the CORI method. The improvement in retrieval effectiveness was a moderate 14.6% at rank position 10, compared to lower baseline. This is caused by the fact that the applied CORI in the experiments firstly weights every sub-collection based on the length of the retrieved documents by each corresponding query, in the entire multilingual collection. Then, it normalizes raw scores of individual lists by making use of these sub-collections' weights such that documents' scores from different sub-collections are either increased or decreased, depending on whether their corresponding sub-collections scores are greater or less than the average score, as described in section 2.3.2. Thus, scores of the documents in the Arabic sub-collection will probably get lower scores while English document scores would be increased, resulting in better performance for the COMB-PCSQ-CORI as English lists are much longer. But, when this run (COMB-PCSQ-CORI) was compared to the best retrieval obtained in the first set of experiments, which was the CRSQM-DECAY-WT-IDF, the former run performed worse than the latter, which still yielded the best results obtained thus far in this chapter. Table 6.7 shows the retrieval effectiveness of the two runs together, whereas Figure 6.11 plots their performance curves, in terms of average DCG.

Measure	Average DCG @									
Run	1	2	3	4	5	6	7	8	9	10
CRSQM-DECAY-WT-IDF	4.255	8.468	11.086	13.096	14.865	16.396	17.578	18.855	20.177	21.336
COMB-PCSQ-CORI	3.40	6.960	9.181	10.821	12.457	13.804	15.257	16.710	17.934	19.066

TABLE 6.7: Retrieval effectiveness, in terms of average DCGs, of the best results obtained by proposed approaches in the centralized architecture (CRSQM-DECAY-WT-IDF) compared to the best results obtained by the proposed methods in the traditional distributed architecture (COMB-PCSQ-CORI).

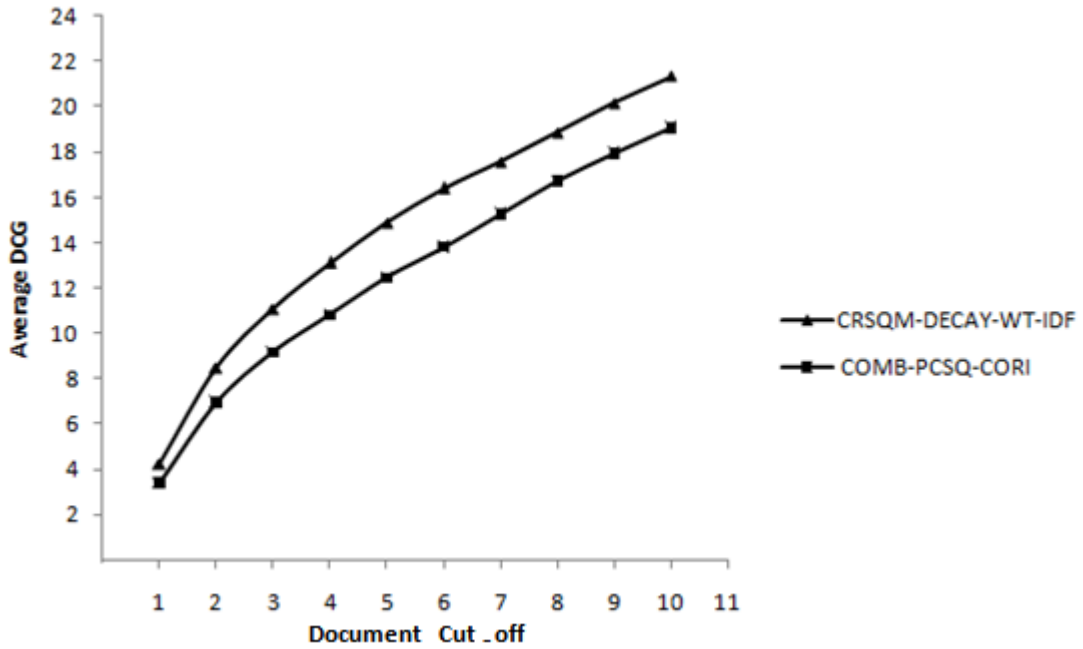


FIG. 6.11: Retrieval effectiveness, in terms of average DCGs, of the best results obtained by proposed approaches in the centralized architecture (CRSQM-DECAY-WT-IDF) compared to the best results obtained by the proposed methods in the traditional distributed architecture (COMB-PCSQ-CORI).

So, why the CRSQM-DECAY-WT-IDF run performs statistically better than the COMB-PCSQ-CORI (p-value = 0.00005) is important, especially with the absence of the overweighting problem in the latter approach, which was basically distributed.

This due to the different length of the utilized query sets. Big and mixed merged queries are often generated for the centralized index. Thus, in the centralized and mixed sub-collection in the combined architecture of indexing, the use of such queries would probably result in high scores for the mixed documents. Thus, when document scores in each corresponding sub-collection in the COMB-PCSQ-CORI run were increased or decreased according to their retrieved documents' length, as in the used CORI, document scores in the inserted centralized and mixed sub-collection in the combined architecture were still high, for some of the queries, although their scores were reduced as this index often resulted in a retrieval of a relatively low number of documents.

In contrast, the CRSQM-DECAY-WT-IDF minimizes the overweighting problem with the use of both cross-lingual structuring strategy and weighted IDF values for terms. Thus, it suppressed the impact of its long queries, especially by the cross-lingual structuring. Thus, the performance of the COMB-PCSQ-CORI was less effective than the CRSQM-DECAY-WT-IDF run. This finding harmonized with those obtained by Rasolofo et al., (2001), who showed the difficulty to yield the similar effectiveness of a centralized architecture by using a distributed technique with a merging strategy.

6.3 Summary

In this chapter, it was shown that most current search-engine-like systems and CLIR systems perform poorly when handling mixed querying because, in most cases, they fail to produce the most relevant documents on top. This is because of the underlying assumption that a CLIR process is a monolingual retrieval that is often preceded by a translation for a monolingual query in the user's native language. The dependence on this principal assumption constrains the majority of the similarity ranking methods in CLIR to be solely based on monolingual weighting and retrieval. Furthermore, most approaches are based on exact matching between queries and documents, which in turn results in the majority of the top returned documents being mixed and containing terms that exactly match the query terms, regardless of the ingredient languages of these queries or those in documents. This suggested that there is a real need for a mixed-language IR system that should be a language-aware system.

It was seen in the chapter that, with an Arabic-English test collection in the computer science domain, the retrieval performance of the CLIR lower baseline experiment, which combines the centralized approach of indexing with the structured query model(s) for weighting, was relatively poor as issues like biased term frequency, overweighting (whether it is monolingual, cross-lingual or caused by mixture of texts) and independent computations of terms that are similar but in different languages, could hurt retrieval significantly. In spite of these problems the effectiveness of the lower baseline retrieval is still better than using the raw mixed queries (with no translation mechanism) to retrieve documents, as it happens in systems that mimic existing search engines.

The alternative for moderating such difficulties in a centralized architecture for indexing, is to use a mixed-language IR system, in which terms that are similar across languages, especially that are technical, are handled as a set of cross-lingual synonymous terms (cross-lingual structured query model). This would reduce the effect of most drawbacks that stemmed from the use of mixed and multilingual documents in a centralized index. Thus, in the CLIR experiments it was shown that the retrieval effectiveness of using a cross-lingual synonymy mechanism for structuring technical terms (which are the most significant keywords), regardless of their languages, was better than using the lower baseline approach and it achieved a comparable efficiency to a monolingual baseline that is based on manually translated queries by experts.

In the experiments, it was also shown that a remarkable improvement over both the lower baseline and the cross-lingual structured query model was observed when the latter approach was strengthened with a diminishing factor for the term frequency and the document length components in those terms, which tend to co-occur with their translations in another language. Co-occurrence of bilingual terms is a major characteristic in non-English documents that cannot be simply ignored when weighting documents in a mixed-language IR system. Thus, the proposed cross-lingual structured query model, which considers bilingual term weights, affected retrieval significantly, especially on the top ranked documents.

For terms in a mixed query, using a weighted inverse document frequency that is based on a damping factor derived from the sizes of sub-collections/languages with regards to the size of the entire collection, could have significant better results over the lower CLIR baseline because the effect of over-weighting

would be minimized. However, results of experiments showed that the impact of using a cross-lingual structuring (cross-lingual structured query model) on overweighted terms is much greater than the impact of utilizing a weighted inverse document frequency approach, while the combination of the two approaches together resulted in a more robust approach that yielded a significant performance improvement over the baseline and acieved 97.507% of the monolingual baseline performance.

In a conventional distributed architecture, the mixed-language problem in the experiments was firstly controlled by using a hybrid architecture of both the centralized and the distributed architectures (combined architecture of indexing) so as to minimize their drawbacks, while at the same time merging their strengths. Then, to overcome the overweighting, which is caused by the mixture of texts, a probabilistic cross-lingual structuring for technical terms can be utilized (probabilistic cross-lingual structured query model). However, the type of the used merging method for merging individual lists is a major component that could have a significant impact on retrieval, despite the use of the proposed approaches (the hybrid architecture for indexing with the probabilistic cross-lingual structured query model). Accordingly, in the experiments, it was shown that the proposed approaches with a merging strategy that is based on raw merging of scores of documents in individual list, or a merging method based on sub-collections statistics, like the CORI, could outperform the lower baseline experiments. Nevertheless, using the proposed approaches with merging methods like those that normalize scores of documents by the maximum score in each individual list could hurt retrieval significantly. In the experiments, such approaches resulted in a similar, but not better, performance to the lower baseline.

Experiments also revealed that the best retrieval results were obtained by using a combination of the cross-lingual structured query model with weighted inverse document frequency, which was essentially based on a centralized approach. In particular, the latter centralized-based approach yielded statistically and significantly better results than the combination of the hybrid architecture of indexing with a probabilistic cross-lingual structured query model, which was distributed.

CHAPTER

7

Conclusion

Non-English-speaking users, such as Arabic speakers, are not able to express terminology in their native languages. Besides the fact that mixing languages together is a natural human tendency, issues like limited modern vocabulary, irregular translation and/or transliteration processes of newly added terms, dominance of English terminology and regional variations, as in the Arabic language, are major reasons for this mixed trend in both queries and documents. Therefore, such non-English-speaking users may express their queries in a mixed form between two languages, mostly English and the native language, in order to precisely present their concepts to search engines.

Current search engines and traditional CLIR and MLIR systems do not handle mixed-language queries and documents adequately. This is because the majority of algorithms, and also test collections, are optimized for monolingual queries, even if they are translated. This underlying monolingual assumption is caused by the fact that in most cases it is presumed that the CLIR and MLIR tasks are primarily reduced to a monolingual retrieval preceded by a translation process.

Inspired by these insights, the major purpose of this work was to experimentally introduce and contribute to the development of mixed-language IR systems and to improve retrieval of mixed-language queries. The mixed-language problem in this thesis has been studied through a corpus that had been created for this purpose. The corpus, which was statistically tested, is multilingual and mixed in both Arabic and English, synchronic and specialized in common computer science vocabulary.

To meet the primary goal of building language-aware algorithms, the main focus of the thesis was to explore weighting components when all documents are placed together into a unified index (centralized architecture) and both weighting and indexing components of IR systems whenever several distributed indices (sub-collections divided according to languages) are used (distributed architecture).

For the weighting components, a cross-lingual re-weighting method (cross-lingual structured query model) was proposed. Thus, for any technical source query term, regardless of its language, it can be

language-aware by obtaining its translations firstly and then all the candidate translations are grouped together with the source term itself, resulting in cross-lingual synonyms. This was done while taking into consideration different forms in which texts in different languages are mixed, e.g., bilingual co-occurring terms in mixed documents, and their impact on retrieval performance. Thus, based on such a mixed-language feature, term frequency, document frequency and document length components were re-estimated using the cross-lingual re-weighted model.

In a centralized architecture, however, another type of re-weighting was also proposed, that is the re-weighted inverse document frequency, which can be performed in different ways. The idea behind re-weighted IDF is based on the assumption that a sub-collection with higher number of documents should contribute more than another sub-collection with small size. Thus, using some sub-collection statistics, a damping factor is computed and then incorporated in DF and/or IDF of terms, depending on the sub-collection to which they belong. Re-weighted inverse document frequency was shown to moderate traditional overweighting of terms that belong to small collections.

The two types of the proposed re-weighting (cross-lingual structured model and re-weighted inverse document frequency), however, can be combined together to improve mixed-language retrieval. Such combination can be done in a sequential manner, meaning to apply a cross-lingual re-weighting followed by a re-weighted IDF. On one hand, the re-weighted IDF could moderate traditional overweighting, which always occurs due to incomparable sizes of sub-collections in a single index. On the other hand, the cross-lingual re-weighting could handle most problems that may occur due to mixture of documents like dominance of mixed documents and overweighting due to a mixture of texts and biased TF and DF, but it causes technical terms to have higher joint document frequencies (low weights), due to cross-lingual structuring, while non-technical terms do not. Thus, a combination between the two approaches could be beneficial.

In a traditional distributed environment, a new architecture that is capable of indexing mixed documents while preserving their contents (and thus their scores are computed from complete mixed documents) was also developed. This is achieved by combining the advantages of the centralized and distributed architectures for MLIR, while trying to moderate their drawbacks. A similar developed cross-lingual weighting to that proposed in the centralized architecture, but in terms of a probabilistic framework, is utilized in this new architecture.

Through the evaluation, with experiments on Arabic and English, the following results were obtained:

a) In a centralized index:
   (1) Current search engines and CLIR systems cannot handle mixed-language queries adequately. This is because, in most cases, their result lists are dominated by mixed-language documents as these approaches tend to perform exact matching between queries and documents, regardless of the languages present in them, rather than retrieving the most relevant documents. Accordingly, there may be many monolingual highly relevant documents that are ranked at the lower level of the ranking. Thus, the result list is biased towards mixed documents.
   (2) Cross-lingual re-weighting (cross-lingual structured query model) could yield statistically better results compared to traditional approaches. Furthermore, it could handle mixed-language

features in queries and documents well as it makes document scores comparable. This is mainly caused by the fact the proposed cross-lingual structured model could suppress the impact of the independent computations of terms that are similar across languages. This is important to mixed-language queries and documents. Independent computation of cross-lingual terms is the major tendency of the majority of the current CLIR and MLIR approaches. The same Argument also applied to document frequency of terms, in which independent computations could result in that one of cross-lingual similar terms may skew the effect of its equivalent counterpart (s) in the other language. Thus, the cross-lingual structuring of terms moderates most these impacts. Additionally, it is found in the experiments that the assumption of not using estimated probabilities for translations is valid because in technical Arabic domain, technical terms that appear as superfluous, due to regional variation in the language, may be placed in highly relevant documents.

(3) Adjustment of term frequencies of co-occurring terms in different languages could affect retrieval of mixed-language documents significantly. In particular, since co-occurring terms usually increase the weights of mixed document in which they occur, using a decaying factor that is based on the number of co-occurrences of such bilingual terms could result in re-ranking result list more accurately. This is especially true if the fact that such co-occurring of terms phenomenon is very prevalent in non-English documents, e.g. Arabic and Chinese.

(4) Re-weighted IDF could minimize overweighting in multilingual document collections, but in multilingual and mixed collections overweighting due to mixture of text has an impact larger than the traditional overweighting effect, at least in terms of the collected corpus. In spite of this fact, re-weighted IDF improves mixed-language querying because in most cases it suppresses the impact of the overweighted terms due to incomparable sizes of sub-collections. Such re-weighting of IDFs would have an important effect on the Web if the fact that English sub-collection is much larger (and thus high document frequencies and low weights for the English terms) when compared with the other non-English languages (sub-collections).

(5) Combination of cross-lingual re-weighting model and re-weighted IDF could yield the best results. It could yield a comparable efficiency to monolingual retrieval using monolingual English queries.

b) In a traditional distributed indexing approach:

(1) It is beneficial to use a combined approach of centralized and distributed architecture whenever a mixed and a multilingual document collection is used. In such an architecture, most drawbacks in these two types of indexing are minimized. The combined architecture is a novel solution because it artificially avoids partitioning mixed documents across different sub-collections, as the traditional distributed architecture does, by creating a centralized index, while at the same time it avoids the overweighting by placing different monolingual indices with each index corresponds to a monolingual sub-collection in a particular language. In that way, the architecture could be used for any types of documents (mixed or monolingual) and any number

of sub-collections (languages) as well. This is important to the IR task due to multilingual feature of the Web.

(2) The use of a unified index for indexing a mixed and multilingual collection is more beneficial than the use of a traditional distributed architecture or the hybrid approach architecture. However, the relatively less performance of the traditional distributed approach is mainly caused by the merging methods which really need enhancements. Although the distributed information retrieval field goes in this direction, most approaches depend on the efficiency of the distributed servers and the bandwidth of the networks so as to download some sample documents to estimate final result scores in different lists, for example. But, in developing countries, such difficulties would probably limit the use of such distributed approaches. Therefore, the conclusion about the use of a single index is important because it would help in any future extension of the language-aware IR systems, which are mainly needed in developing non-English countries.

These results lead to the conclusion that indeed it is possible to develop an IR system that can handle mixed queries and mixed documents effectively. There are many problems introduced by the explicit handling of multiple languages but the algorithms and experiments conducted demonstrate that these problems can be adequately resolved in an IR system. The evidence suggests that language-awareness and mixed-language solutions are feasible for IR systems without diminishing quality of results.

With information globalization and moving towards an international community, it becomes essential to not constrain non-English speakers, such as Arabic users to single languages. The algorithms proposed in this thesis address the Web searching needs of such non-English speakers, who often need the most relevant information rather than just retrieving documents contain exactly their queries terms. The proposed algorithms could empower and present a direction for future search engines, which should allow multilingual users (and their multilingual queries) to retrieve relevant information created by other multilingual users. Thus, it could have significant outcomes for languages with limited modern vocabulary, mostly those non-English ones, in developing countries. In that context, many non-English users would be better serve when they need to search relevant information of their information needs.

Limitations

Although the proposed approaches showed significant improvement for mixed-language IR systems, there are some major limitations. First of all, the proposed approaches, specifically the re-weighting parts, depend solely on translations obtained from specialized dictionaries. Such dictionaries are not always available for many languages. Additionally, dictionaries would not cover every term in the vocabulary, resulting in an OOV problem. This is especially true in scientific domains, e.g., computer science, whose vocabularies are always evolving and the lack of up-to-date terminology could have a significant impact on retrieval. Such a limitation would hinder efficiency of proposed approaches and render them to a conventional centralized architecture. However, the OOV problem is not limited to proposed approaches in this thesis but it can occur in most approaches in CLIR and MLIR. Apparently, to

increase the efficiency of proposed solutions, a better translation approach is needed, e.g., the use of the Web Such an approach would minimize OOV terms.

Another major limitation is that the proposed re-weighting approaches (cross-lingual re-weighting and re-weighted IDF) did not make use of language identification and detection techniques. In the experiment, a simple language identifier was used. But, ideally, a complete approach for identifying language for each term/portion/paragraph in both mixed documents and queries is needed before even indexing documents and/or matching queries with documents. Such an approach should be able also to identify mixed documents from monolingual ones. Language identification is essential when the proposed approaches are used with multilingual and mixed document collections with many languages, instead of only bilingual. In such a case, queries can be written in different bilingual mixed queries and, thus, unless an accurate language identifier and/or classifier, e.g., a statistical language model using hidden Markov models for estimating probabilities of sentences, is used, wrong stemmers and/or documents may be selected and retrieved. Furthermore, any built language model should be trained on scientific data as the likelihood of similarities in letters will increase, for example, both Arabic and Persian are written in Arabic script but, as technical terms in both languages are borrowed from English, the process of language detection would be a little difficult.

Another limitation to the work presented, is the corpus. Ideally, a standard test collection should be used in experiments so as to allow accurate repeatability of experiments. However, the data set used in experiments is not a standard because of the reasons provided earlier.

Another limitation is the fact that the proposed approaches are computationally costly. This is especially true when each co-occurrence of bilingual terms in each document is to be tested regardless of order. This makes response time of proposed approaches for mixed-language IR relatively high. Nevertheless, such overhead can be transferred to indexing time (Levow, et al, 2005)

For the proposed architecture, the major limitation is that it relies on the use of the conventional merging methods with the same IR model. It was shown that most of these methods, except CORI, have some drawbacks and thus it could have negative impact on retrieval performance. Accordingly, it is better to develop other merging methods that could incorporate the mixed-language feature in the proposed architecture.

CHAPTER

8 Future Work

A number of potential directions are worthy to be explored in the future. However, it is firstly being planned to extend the size of the MULMIXEAC corpus. The targeted size is 200,000 documents and the planned sources from which the corpus would be extended are the electrical and electronic domains, as those ones are close to computer science. The same Arabic and English languages are still the focus for this extension. During this stage also, a mixed language identifier will be developed. It was shown in the limitation that language identifiers are important to mixed-language IR system. The focus, however, becomes on Arabic script, which is used by other languages such as Urdu and Persian, and on mixed documents of such non-English languages with English. Once the corpus is expanded, other investigations for mixed-language feature would be considered. The following sub-sections illustrate the future directions.

8.1 Phrase-Based Structuring and Web-based Translation

Investigation firstly would be focused on the implementation of other techniques for OOV terms. For example, those making use of bilingual search-results snippets (query-based summary) in mixed documents and/or the use of hyperlinks and anchor texts. This would probably minimize OOV problem and increase the effectiveness of the proposed solutions. The use of probabilistic-based knowledge in translation can be also tested to check whether it is beneficial to implement a probabilistic-based translation approach. However, such an approach needs some collaboration firstly with publishers to acquire several translation sources and/or references, especially those are translated in both Arabic and English (parallel), in order to estimate possible translations for scientific terms.

Following this, the future direction would be focus on applying the proposed mixed-language approaches on phrase-based translations and structuring and whether such an approach could have substantial

impacts on the proposed approaches ("[Douglas Oard, personal communications, 2012]"). It is observed that English portions, which are composed from more than one word, in mixed queries are mostly phrases.

So, instead of using term-based translation and/or term-based query structuring, the next set of experiments would be focused on exploring the impacts of phrase-based translation, in which phrase translation is performed as a single unit. On one hand, it was shown that phrase-based translation approach, as a first option for performing translation followed by a word-based translation - if the phrase is not found, could yield significant improvement on results (Ballesteros and Croft, 1998, for example). But, it was also shown that the major problem with phrase translation is that they are not always found in dictionaries. During the experiments of this thesis, however, it was noted that there is a higher likelihood that the phrase can be found in scientific dictionaries than a single word that participates in composing a phrase.

Note that the need for a POS tagger to identify phrases in mixed queries would be minimized as consecutive words in English portions in such queries are more likely to be phrases.

On the other hand, it was shown that the term-based structuring, is more effective (Pirkola, et al., 2001; Pirkola, et al., 2002). However, the experiments of Pirkola were applied with the underlying assumption that a CLIR is a translation followed by a monolingual retrieval, meaning that the phrase-based query structuring was monolingual. Thus, it is interesting to determine which approach is better when moving to proposed cross-lingual weighting (cross-lingual phrase-based structuring).

8.2 Results Merging Methods

Results merging is another potential direction for future work. With the growing interest in distributed architectures, the question in which the study needs further investigation is how to merge results when both mixed and monolingual documents are indexed in a modified distributed architecture, as in the proposed hybrid indexing approach. In particular, it was shown that this proposed indexing architecture is capable of indexing documents, regardless of languages, but the merging problem is different and not being explored in mixed-language problems.

Therefore, it is planned to explore merging methods, specifically logistic regression models by incorporating a mixed-language parameter. Until now there is no logistic regression model that incorporates the parameter of how much a document is mixed, which can be converted to a probability, although logistic regression is well studied. Thus, instead of using only original document scores and their ranks to predict probabilities of relevance of documents when documents are merged, three coefficients/parameters will be employed for fitting a logistic regression model in the proposed hybrid architecture. In such a case the MULMIXEAC will be employed for training the model.

8.3 Field Weightings of Mixed Documents in BM25

Another direction for future investigation is the BM25 Okapi field weighting space ("[Douglas Oard, personal communications, 2011]"). It was shown that some approaches extended the model to multiple weighted fields. Furthermore, it was described how fields in documents in this extended model are weighted and how scores obtained from different fields in documents are combined. Such simple extension to multiple weighted fields was shown to be effective, yet fields of documents are still in a monolingual language. In particular, it is aimed to explore whether an estimated probability of how much the document is mixed can be incorporated in field weights (note that fields are in several languages).

8.4 Co-occurrence Measures

The above point also suggests the possibility of extending other weighting schemes, to handle multilinguality, in terms of co-occurrence measures. If MULMIXEAC is used, for example, as a target collection, then it is possible to produce a matrix of co-occurrence values between each pair of terms in two languages (cross-lingual co-occurrence measure instead of monolingual co-occurrence). Next, it may be interesting to test whether such co-occurrence values can be incorporated in term weights in some way such that terms with high co-occurrence association values will reduce documents in which they occur, whereas infrequent co-occurred terms will reduce their corresponding documents less. Such an approach will transfer ad-hoc computation of proposed cross-lingual re-weighting to earlier stages when building the term co-occurrence matrix.

8.5 Arabic Regional Variation in Scientific Domain

Another potential motivation to carry out more thorough investigation is the problem of Arabic regional variation in scientific domains. Contrary to general domains, the problem is more challenging in scientific domains because there is a large divergence in vocabulary across the big Arabic-speaking region, unlike in general-purpose vocabulary. Nevertheless, the process may be easier because it is possible to target regional variants that co-occur with English terms. In such a case, the English terms are used to collect regional variations. Thus, if statistical language models for different regions are incorporated, then this may help in grouping regional variant words into semantic classes and, thus, improving retrieval of mixed-language IR systems.

Although the proposed approaches in this thesis were focused on Arabic-English languages, it is interesting to show also their impacts on other languages, for example Japanese and English in NTCIR collections ("[Douglas Oard, personal communications, 2011]"), so as to investigate consistency. This would probably examine if the proposed mixed-language solutions are fit for any language pair and would investigate their consistency.

Domains like computer science are usually rich in acronyms. In this thesis, this feature was not explored. Therefore, in the future work, acronym expansion methodology should be tested. On one hand, expansion of acronyms seems reasonable but it is not clear whether such expansion would hurt retrieval or not.

Bibliography

Abdelali, A. 2006. *Improving Arabic information retrieval using local variations in modern standard Arabic*. *PhD Thesis*. New Mexico Institute of Mining and Technology.

Abdelali, A., Cowie, J. & Soliman, H. 2005. Building a modern standard Arabic corpus. *In: workshop on computational modeling of lexical acquisition*. The split meeting. Croatia, 25th to 28th of july, 2005. Citeseer.

Abdelali, A., Cowie, J. & Soliman, H. S. 2007. Improving query precision using semantic expansion. *Information processing & management*, 43, 705-716.

Abduljaleel, N. & Larkey, L. 2003a. English to Arabic transliteration for information retrieval: A statistical approach. *Center for Intelligent Information Retrieval Computer Science*, University of Massachusetts.

Abduljaleel, N. & Larkey, L. 2003. Statistical transliteration for English-Arabic cross language information retrieval. *In: Proceedings of the twelfth international conference on Information and knowledge management*, 2003b. ACM, 139-146.

Abusalah, M., Tait, J. & Oakes, M. 2005. Literature review of cross-language information retrieval. *In: Transactions on Engineering, Computing and Technology*, ISSN, 2005. Citeseer.

Adriani, M. & Van rijsbergen, C. 2000. Phrase identification in cross-language information retrieval. *In: Proceedings of the RIAO (Recherche d'Informations Assistée par Ordinateur) Conference: Content-Based Multimedia Information Access (RIAO 2000)*, 2000. Citeseer, 520-528.

AL-Shammari, E. & Lin, J. 2008. A novel Arabic lemmatization algorithm. *In: Proceedings of the second workshop on Analytics for noisy unstructured text data*, 2008a. ACM, 113-118.

Al-shammari, E. T. & Lin, J. 2008. Towards an error-free Arabic stemming. *In: Proceedings of the 2nd ACM workshop on Improving non english web searching*, 2008b. ACM, 9-16.

Alansary, S., Nagi, M. & Adly, N. 2007. Building an International Corpus of Arabic (ICA): progress of compilation stage. *In: 7th International Conference on Language Engineering, Cairo, Egypt*, 5–6 December 2007, 2007.

Alansary, S., Nagi, M. & Adly, N. 2008. Towards Analyzing the International Corpus of Arabic (ICA): Progress of Morphological Stage. *In: 8th International Conference on Language Engineering, Egypt*, 2008.

Aljlayl, M. & Frieder, O. 2001. Effective arabic-english cross-language information retrieval via machine-readable dictionaries and machine translation. *In: Proceedings of the tenth international conference on Information and knowledge management, 2001*. ACM, 295-302.

Aljlayl, M. & Frieder, O. 2002. On Arabic search: Improving the retrieval effectiveness via light stemming approach. *In: Proceedings of the 11th ACM International Conference on Information and Knowledge Management, 2002 Illinois*. ACM Press, 340–347.

Aljlayl, M. & Frieder, O. & Grossman, D. 2002. On bidirectional English–Arabic search. *Journal of the American Society for Information Science and Technology*, 53, 1139-1151.

Alkhfajy, S. 1292. Hashiyat El Shihab Ala Tefseer Albiudawy, Dar sader,Buriet.416.

Alotaiby, F., Alkharashi, I. & Foda, S. 2009. processing large arabic text corpora: preliminary analysis and results. *In: Proceedings of the 2nd international conference on Arabic Langauge Resources and Tools, 2009 Cairo, Egypt*. 22-23.

Attia, M. A. 2007. Arabic tokenization system. *In: Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, 2007. Association for Computational Linguistics, 65-72.

Attia, M. A. 2008. Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation. *PhD Thesis*. The University of Manchester.

Ballesteros, L. & Croft, W. B. 1997. Phrasal translation and query expansion techniques for cross-language information retrieval. *In: ACM SIGIR Forum*, 1997. ACM, 84-91.

Ballesteros, L. & Croft, W. B. 1998. Resolving ambiguity for cross-language retrieval. *In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998. ACM, 64-71.

Baroni, M. & AND Bernardini, S. 2006. Bologna: Gedit.

Belkin, N. J. & Croft, W. B. 1992. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35, 29-38.

Benajiba, Y. & Rosso, P. 2007. Towards a Measure for Arabic Corpora Quality. *Proceedings of International Colloquium on Arabic Language Prcessing*. 2007. CITALA.

Boughanem, M., Chrisment, C. & Nassr, N. 2002. Investigation on disambiguation in CLIR: Aligned corpus and bi-directional translation-based strategies. *In: Evaluation of Cross-Language Information Retrieval Systems*, 2002. Springer, 1-16.

Broglio, J., Callan, J. P. & Croft, W. B. 1994. Inquery system overview. *In: Proceedings of a workshop on held at Fredericksburg, Virginia*: September 19-23, 1993, 1994. Association for Computational Linguistics, 47-67.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D. & Mercer, R. L. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19, 263-311.

Buckley, C. 2000. The TREC-9 Query Track. eds. Voorhees, E. M. and Harman, D., *In: Proceedings of the 9th Text REtrieval Conference (TREC-9)*. 2000. 81-85.

Buckley, C., Salton, G., Allan, J. & Singhal, A. 1995. Automatic query expansion using SMART: TREC 3. *NIST special publication SP*, 69-80.

Buckley, C., walz, J., Mitra, M. & Cardie, C. 1998. Using clustering and superconcepts within SMART: TREC 6. *NIST special publication SP*, 107-124.

Buckwalter, T. 2002. Buckwalter Arabic morphological analyzer version 1.0. *Linguistic Data Consortium*, University of Pennsylvania.

Buckwalter, T. 2004. Issues in Arabic orthography and morphology analysis. *In: Proceedings of the COLING 2004 Workshop on computational approaches to Arabic script-based languages*, 2004. 31-34.

Callan, J. P., Lu, Z. & Croft, W. B. 1995. Searching distributed collections with inference networks. *In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval,* 1995. ACM, 21-28.

Chen, A. 2002. Multilingual information retrieval using english and chinese queries. *In: Evaluation of cross-language information retrieval systems*, 2002. Springer, 373-381.

Chen, A. & Gey, F. 2002. Building an Arabic stemmer for information retrieval. *In: Proceedings of TREC 2002*, 2002.

Chen, A. & Gey, F. 2004a. Combining query translation and document translation in cross-language retrieval. Comparative Evaluation of Multilingual Information Access Systems, 121-124.

Chen, A. & Gey, F. C. 2004b. Multilingual information retrieval using machine translation, relevance feedback and decompounding. *Information Retrieval*, 7, 149-182.

Cheng, J., Y, P., W, L. & L, C. 2004. Creating Multilingual Translation Lexicons with Regional Variations Using Web Corpora. *In: Proceedings of the ACL* 2004a. ACM, 535-542.

Cheng, P. J., Teng, J. W., Chen, R. C., Wang, J. H., Lu, W. H. & Chien, L. F. 2004. Translating unknown queries with web corpora for cross-language information retrieval. *In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004b. ACM, 146-153.

Cheung, P. & Fung, P. 2004. Translation disambiguation in mixed language queries. Machine translation, 18, 251-273.

Cheung, W. 2008. Web searching in a multilingual world. *Communications of the ACM*, 51, 32-40.

Cheung, W., Bonillas, A., Lai, G., XI, W. & Chen, H. 2006. Supporting non-English Web searching: An experiment on the Spanish business and the Arabic medical intelligence portals. *Decision support systems*, 42, 1697-1714.

Church, K. W. & Hanks, P. 1990. Word association norms, mutual information, and lexicography. Computational Linguistics, 16, 22-29.

Cleverdon, C. W. 1962. Report on the Testing and Analysis of an Investigation Into the Comparative Efficiency of Indexing Systems: an investigation supported by a grant to ASLIB by the National Science Foundation, sn].

Croft, W. B., Metzler, D. & Strohman, T. 2010. Search engines: Information retrieval in practice, Addison-Wesley.

Daoud, D. & Hasan, Q. 2011. Stemming arabic using longest-match and dynamic normalization. *In: the proceedings of the 3rd Arabic language Technology International Conference (ALTIC)* -2011. Alexandria, Egypt.

Darwish, K. 2002. Building a shallow Arabic morphological analyzer in one day. *In: Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2002a. 1-8.

Darwish, K. 2002b. *Al-stem: A light Arabic Stemmer*.

Darwish, K. & Oard, D. W. 2003. Probabilistic structured query methods. *In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 2003a. ACM, 338-344.

Darwish, K. & Oard, D. W. 2003b. CLIR Experiments at Maryland for TREC-2002: Evidence combination for Arabic-English retrieval. *In: TREC 2003 proceedings*.

Davis, M. W. 1998. On the effective use of large parallel corpora in cross-language text retrieval. *Cross-language information retrieval*, 11-22.

Deyoung, T. 1999. Arabic language history. [Online]. Available: http://www.indiana.edu/~arabic/arabic_history.htm [Accessed 5/2/ 2013].

Diab, M., Hacioglu, K. & Jurafsky, D. 2004. Automatic tagging of Arabic text: From raw text to base phrase chunks. *In: Proceedings of HLT-NAACL 2004: Short Papers*, 2004. Association for Computational Linguistics, 149-152.

Dunlop, M. D. & Van rijsbergen, C. 1993. Hypermedia and free text retrieval. *Information processing & management,* 29, 287-298.

Fox, E. 1984. Extending the Boolean and vector space models of information retrieval with p-norm queries and multiple concept types. PhD Thesis. Cornell University.

Fraser, A., Xu, J. & Weischedel, R. 2002. TREC 2002 Cross-lingual Retrieval at BBN. *In: TREC 2002 proceedings*, 2002.

Fung, P., Xiaohu, L. & Shun, C. C. 1999. Mixed language query disambiguation. *In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999. Association for Computational Linguistics, 333-340.

Gale, W. A. & Church, K. W. 1991. A program for aligning sentences in bilingual corpora. *In: Proceedings of the 29th annual meeting on Association for Computational Linguistics*, 1991. Association for Computational Linguistics, 177-184.

Gao, J. & Nie, J. Y., Xun, E., Zhang, J., Zhou, M. & Huang, C. 2001. Improving query translation for cross-language information retrieval using statistical models. *In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001. ACM, 96-104.

Gao, J. & Nie, J. Y., Zhang, J., Xun, E., Su, Y., Zhou, M. & Huang, C. 2000. Trec-9 CLIR experiments at MSRCN. *In: The Ninth Text Retrieval Conference (TREC-9)*, 2000. 343–353.

Gao, J., Zhou, M., Nie, J. Y., He, H. & Chen, W. 2002. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. *In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002. ACM, 183-190.

Garcia-molina, S. R. H. & Raghavan, S. 2001. Crawling the Hidden Web. *In: 27th International Conference on Very Large Data Bases*, 2001.

Gey, F., Jiang, H., Petras, V. & Chen, A. 2001. Cross-language retrieval for the CLEF collections—comparing multiple methods of retrieval. *Cross-Language Information Retrieval and Evaluation*, 116-128.

Gey, F. C., Kando, N. & Peters, C. 2005. Cross-language information retrieval: the way ahead. *Information processing & management*, 41, 415-431.

Gey, F. C. & Oard, D. W. 2001. The TREC-2001 cross-language information retrieval track: Searching Arabic using English, French or Arabic queries. *In: TREC 2001*, 2001. 16-26.

Goweder, A. & De roeck, A. 2001. assessment of significant Arabic corpus. *In: Arabic NLP workshop at ACL/EACL*, 2001.

Goweder, A., Poesio, M. & De roeck, A. 2004. Broken plural detection for arabic information retrieval. *In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004. ACM, 566-567.

Goweder, A., Poesio, M. & De roeck, A. & Reynolds, J. 2005. Identifying broken plurals in unvowelised Arabic text. *In: Proceedings of EMNLP*, 2005. 246-253.

Graff, D. 2007. Arabic Gigaword Third Edition .*Linguistic Data Consortium*. Philadelphia, USA. .

Graff, D., Kong, J., Chen, K. & Maeda, K. 2007. English Gigaword Third Edition. *Linguistic Data Consortium*. Philadelphia,USA.

Graff, D. & Walker, K. 2001. Arabic Newswire Part 1. *Linguistic Data Consortium*. Philadelphia,USA. .

Grefenstette, G. 1998. Cross-language information retrieval, *Kluwer* Academic Publishers.

Grossman, D. A. & Frieder, O. 2004. Information retrieval: Algorithms and heuristics, Springer.

Habash, N. & Rambow, O. 2006. MAGEAD: a morphological analyzer and generator for the Arabic dialects. *In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006. Association for Computational Linguistics, 681-688.

Habash, N. & Rambow, O. 2007. Arabic diacritization through full morphological tagging. *In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*; Companion Volume, Short Papers, 2007. Association for Computational Linguistics, 53-56.

Hammo, B. H. 2009. Towards enhancing retrieval effectiveness of search engines for diacritisized Arabic documents. *Information Retrieval*, 12, 300-323.

Hansen, P., Petrelli, D., Karlgren, J., Beaulieu, M. & Sanderson, M. 2002. User-centered interface design for cross-language information retrieval. *In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002. ACM, 383-384.

Hegazi, N. & El-sharkawi, A. 1985. An approach to a computerized lexical analyzer for natural Arabic text. *In: Proceedings of the Arabic Language Conference*, Kuwait, 1985.

Hiemstra, D. 2000. Using language models for information retrieval. CTIT *Ph.D. thesis*.

Hiemstra, D., Kraaij, W., Pohlmann, R. & Westerveld, T. 2001. Translation resources, merging strategies, and relevance feedback for cross-language information retrieval. *Cross-Language Information Retrieval and Evaluation*, 102-115.

Hmeidi, I., Kanaan, G. & Evens, M. 1998. Design and implementation of automatic indexing for information retrieval with Arabic documents. *Journal of the American Society for Information science*, 48, 867-881.

Hull, D. A. & Grefenstette, G. 1996. Querying across languages: a dictionary-based approach to multilingual information retrieval. *In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 1996. ACM, 49-57.

Jackson, P. & Moulinier, I. 2007. Natural language processing for online applications: Text retrieval, extraction and categorization, John Benjamins Publishing Company.

Jang, M. G., Myaeng, S. H. & PARK, S. Y. 1999. Using mutual information to resolve query translation ambiguities and query term weighting. *In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999. Association for Computational Linguistics, 223-229.

Järvelin, K. & Kekäläinen, J. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20, 422-446.

Jones, S. 1975. Report on the need for and provision of an" ideal" information retrieval test collection.

Kadri, Y. & Nie, J. Y. 2006. Effective stemming for Arabic information retrieval. *In: proceedings of the Challenge of Arabic for NLP/MT Conference*, Londres, Royaume-Uni, 2006.

Kadri, Y. & Nie, J. Y. 2007. Combining resources with confidence measures for cross language information retrieval. *In: Proceedings of the ACM first Ph. D. workshop in CIKM*, 2007. ACM, 131-138.

Kashani, M. M., Popowich, F. & Sadat, F. 2007. Automatic transliteration of proper nouns from Arabic to English. *In: Proceedings of the Second Workshop on Computational Approaches to Arabic Script-based Languages*, 2007. 275-282.

Kekäläinen, J. 2005. Binary and Graded Relevance in IR evaluations—Comparison of the Effects on Ranking of IR Systems. *Information processing & management*, 41, 1019-1033.

Khalwaih, I. 2012. Lion Names [Online]. Available: http://www.fustat.com/adab/asmaalasad.pdf [Accessed January 2013].

Khoja, S. & Garside, R. 1999. Stemming Arabic Text. Lancaster, UK, Computing Department, Lancaster University.

Kishida, K. 2005. Technical issues of cross-language information retrieval: a review. *Information processing & management*, 41, 433-455.

Kraaij, W., Nie, J. Y. & Simard, M. 2003. Embedding web-based statistical translation models in cross-language information retrieval. *Computational Linguistics*, 29, 381-419.

Kraaij, W. & Pohlmann, R. 1996. Viewing stemming as recall enhancement. *In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 1996. ACM, 40-48.

Krovetz, R. 1993. Viewing morphology as an inference process. *In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, 1993. ACM, 191-202.

Landauer, T. K. & Littman, M. L. 1990. A statistical method for language-independent representation of the topical content of text segments. *In: Proceedings of the Eleventh International Conference: Expert Systems and Their Applications*, 1990. Citeseer, 85.

Larkey, L., Ballesteros, L. & Connell, M. 2007. Light stemming for Arabic information retrieval. *Arabic computational morphology*, 221-243.

Larkey, L. S., Abduljaleel, N. & Connell, M. 2003. What's in a Name?: Proper Names in Arabic Cross Language Information Retrieval. *In: ACL Workshop on Comp. Approaches to Semitic Languages*, 2003. Citeseer.

Larkey, L. S., Ballesteros, L. & Connell, M. E. 2002. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. *In: Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 25 th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002. Citeseer, 275-282.

Larkey, L. S. & Connell, M. E. 2006. Arabic information retrieval at UMass in TREC-10. DTIC Document.

Levow, G. A. & Oard, D. W. 2002. Signal boosting for translingual topic tracking. *Topic detection and tracking*, 175-195.

Levow, G. A. & Oard,D. W. & Cabezas, C. I. 2000. Translingual topic tracking with PRISE. *In: Working Notes of the Third Topic Detection and Tracking Workshop*, 2000.

Levow, G. A. & Oard,D. W. & Resnik, P. 2005. Dictionary-based techniques for cross-language information retrieval. *Information processing & management*, 41, 523-547.

Li, Q., Chen, Y. P., Myaeng, S. H., Jin, Y. & Kang, B. Y. 2009. Concept unification of terms in different languages via web mining for Information Retrieval. *Information processing & management*, 45, 246-262.

Lin, W. C. & Chen, H. H. 2003. Merging mechanisms in multilingual information retrieval. *Advances in Cross-Language Information Retrieval*, 175-186.

Lin, W. C. & Chen, L. F. & Lee, H. J. 2004. Anchor text mining for translation of Web queries: A transitive translation approach. *ACM Transactions on Information Systems* (TOIS), 22, 242-269.

Lu, Y., Chau, M., Fang, X. & Yang, C. C. 2006. Analysis of the Bilingual Queries in a Chinese Web Search Engine. *In: Proceedings of the Fifth Workshop on E-Business* (2006, Milwaukee, Wisconsin, USA), 2006. Citeseer.

Maamouri, M., Bies, A. & Kulick, S. 2006. Diacritization: A challenge to arabic treebank annotation and parsing. *In: Proceedings of the Conference of the Machine Translation SIG of the British Computer Society,* 2006.

Manning, C. D., Raghavan, P. & Schutze, H. 2008. Introduction to information retrieval, Cambridge University Press Cambridge.

Manning, C. D. & Schutze, H. 1999. Foundations of statistical natural language processing, MIT press.

Mansour, N., Haraty, R. A., Daher, W. & Houri, M. 2008. An auto-indexing method for Arabic text. *Information processing & management*, 44, 1538-1545.

Manzour, I. 2009. Lisan Al-Arab. [Online]. Available: www.lesanarab.com [Accessed January 2010].

Maron, M. E. & Kuhns, J. L. 1960. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7, 216-244.

Mccarley, J. S. 1999. Should we translate the documents or the queries in cross-language information retrieval? *In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999. Association for Computational Linguistics, 208-214.

Mccarley, A. 2006. Corpus-based language studies: An advanced resource book, Routledge.

Mccarley,P. & Mayfield, J. 2002. Comparing cross-language query expansion techniques by degrading translation resources. *In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002a. ACM, 159-166.

Mcnamee, P. & Mayfield, J. 2002. JHU/APL experiments at CLEF: Translation resources and score normalization. *In: Evaluation of cross-language information retrieval systems*, 2002b. Springer, 1283-1301.

Miniwatts Marketing Group. 2013. Internet World Stats Usage and Population Statistics [Online]. Available: http://www.internetworldstats.com/stats7.htm [Accessed January 2013].

Mirkin, B. 2010. Population levels, trends and policies in the Arab region: Challenges and opportunities. Arab Human Development, Report Paper, 1.

Mohamed, M., Arafa, W., Darwish, K. & Gheith, M. 2011. Using WIKIPEDIA for Retrieving Arabic Documents. *In: the proceedings of the 3rd Arabic language Technology International Conference (ALTIC)*-2011. Alexandria, Egypt.

Molina-salgado, H., Moulinier, I., Knudson, M., Lund, E. & Sekhon, K. 2002. Thomson legal and regulatory at CLEF 2001: Monolingual and bilingual experiments. *In: Evaluation of cross-language information retrieval systems*, 2002. Springer, 163-177.

Monz, C. & Dorr, B. J. 2005. Iterative translation disambiguation for cross-language information retrieval. *In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005. ACM, 520-527.

Mori, T., Kokubu, T. & Tanaka, T. 2001. Cross-lingual information retrieval based on LSI with multiple word spaces. In: Proceedings of the Second NTCIR Workshop Meeting on Evaluation of Chinese & Japanese Text Retrieval and Text Summarization, 2001.

Moukdad, H. 2006. Stemming and root-based approaches to the retrieval of Arabic documents on the Web. *Webology*, 3.

Mustafa, S. H. & Al- radaideh, Q. A. 2004. Using N- grams for Arabic text searching. *Journal of the American Society for Information Science and Technology*, 55, 1002-1007.

Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41, 10.

Nie, J. Y. 1999. CLIR using a probabilistic translation model based on Web documents. Trec8. http://trec. nist. gov/pubs/trec8/t8_proceedings. html.

Nie, J. Y. 2003. Cross-language information retrieval. *IEEE Computational Intelligence Bulletin*, 2, 19-24.

Nie, J. Y. 2010. Cross-language information retrieval. Synthesis Lectures on Human Language Technologies, 3, 1-125.

Nie, J. Y., isabelle, P., Plamondon, P. & Foster, G. 1998. Using a probabilistic translation model for cross-language information retrieval. *In: Sixth workshop on Very Large Corpora*, 1998. 18-27.

Nie, J. Y. & Jin, F. 2002. Merging different languages in a single document collection. *In: Proceedings of CLEF*, 2002. Citeseer.

Nie, J. Y. & Jin, F. 2003. A Multilingual Approach to Multilingual Information Retrieval. *Advances in Cross-Language Information Retrieval*, 101-110.

Nie, J. Y. & Simard, M. 2001. Using Statistical Translation Models for Bilingual IR. *In: Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, 2001. Springer-Verlag, 137-150.

Nie, J. Y. & Simard, M., Isabelle, P. & Durand, R. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. *In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999. ACM, 74-81.

Oard, D. 1998. A comparative study of query and document translation for cross-language information retrieval. *Machine Translation and the Information Soup*, 472-483.

Oard, D., Levow, G. A. & Cabezas, C. 2000. CLEF experiments at Maryland: Statistical stemming and backoff translation. *Cross-Language Information Retrieval and Evaluation*, 176-187.

Oard, D. W. & Gey, F. 2002. The TREC-2002 Arabic/English CLIR Track. *In: Proceedings of the Eleventh Text REtrieval Conference*, 2002.

Och, F. J. & Ney, H. 2000. Improved statistical alignment models. *In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000. Association for Computational Linguistics, 440-447.

Och, F. J. & Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29, 19-51.

P´erez-Iglesias, J., P´erez-Ag¨uera, J., Fresno, V and Feinstein, Y. 2009. Integrating the Probabilistic Model BM25/BM25F into Lucene, arXiv: 0911.504 CS.IR, 2.

Paice, C. D. 1984. Soft evaluation of Boolean search queries in information retrieval systems. *Information Technology: Research and Development*, 3, 33-41.

Paice, C. D. 1994. An evaluation method for stemming algorithms. *In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1994. Springer-Verlag New York, Inc., 42-50.

Paltoglou, G., Salampasis, M. & Satratzemi, M. 2007. Hybrid results merging. *In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007. ACM, 321-330.

Parker, R., Graff, D., Jumbo, K., Ke, C. & AND Kazuaki, M. 2011. English Gigaword. Fifth Edition. *Linguistic Data Consortium*, Philadelphia.

Petrelli, D., Beaulieu, M., Sanderson, M., Demetriou, G., Herring, P. & Hansen, P. 2004. Observing users, designing clarity: A case study on the user- centered design of a cross- language information retrieval system. *Journal of the American Society for Information Science and Technology*, 55, 923-934.

Pirkola, A. 1998. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. *In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998. ACM, 55-63.

Pirkola, A. 2001. Morphological typology of languages for IR. Journal of Documentation, 57, 330-348.

Pirkola, A., Hedlund, T., Keskustalo, H. & Järvelin, K. 2001. Dictionary-based cross-language information retrieval: problems, methods, and research findings. *Information Retrieval*, 4, 209-230.

Pirkola, A., PuolamäkI, D. & Järvelin, K. 2003. Applying query structuring in cross-language retrieval. *Information processing & management*, 39, 391-402.

Porter, M. 2009. Snowball: A language for stemming algorithms, 2001. URL http://snowball. tartarus. org/texts/introduction. html.

Porter, M. F. 1980. An algorithm for suffix stripping. Program.

Powell, A. L., French, J. C., Callan, J., Connell, M. & Viles, C. L. 2000. The impact of database selection on distributed searching. *In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000. ACM, 232-239.

Pravec, N. 2002. Survey of learner corpora. ICAME, 26, 81-114.

Rasolofo, Y., Abbaci, F. & Savoy, J. 2001. Approaches to collection selection and results merging for distributed information retrieval. *In: Proceedings of the tenth international conference on Information and knowledge management*, 2001. ACM, 191-198.

Resnik, P. 1998. Parallel strands: A preliminary investigation into mining the web for bilingual text. *Machine Translation and the Information Soup*, 72-82.

Resnik, P. & Smith, N. A. 2003. The web as a parallel corpus. Computational Linguistics, 29, 349-380.

Rieh, H. & Rieh, S. Y. 2005. Web searching across languages: Preference and behavior of bilingual academic users in Korea. *Library & information science research*, 27, 249-263.

Robertson, S. & Sparck jones, K. 1976. Relevance Weighting of Search Terms. *Journal of the American Society for Information science*, 27, 129-46.

Robertson, S., Zaragoza, H. & Taylor, M. 2004. Simple BM25 extension to multiple weighted fields. *In: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, 2004. ACM, 42-49.

Robertson, S. E. & Walker, S. 1994. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. *In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, 1994. Springer-Verlag New York, Inc., 232-241.

Rocchio, J. J. 1971. Relevance feedback in information retrieval. 313–323

Rogati, M. & Yang, Y. 2004. Resource selection for domain-specific cross-lingual IR. *In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004. ACM, 154-161.

Saad, M. K. & Ashour, W. 2010. OSAC: Open Source Arabic Corpora. the 6th International Conference on Electrical and Computer Systems (*EECS'10*).

Sakai, T. 2007. On the reliability of information retrieval metrics based on graded relevance. Information processing & management, 43, 531-548.

Salhi, A. & Yahya, A. 2011. Tools for Arabic People Names Processing and Retrieval. *In: the proceedings of the 3rd Arabic language Technology International Conference (ALTIC)*-2011. Alexandria, Egypt.

Salton, G. 1971. The SMART retrieval system—experiments in automatic document processing.

Salton, G. & Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24, 513-523.

Salton, G., Wong, A. & Yang, C. S. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18, 613-620.

Samy, D., Sandoval, A. M., Guirao, J. M. & Alfonseca, E. 2006. Building a Parallel Multilingual Corpus (Arabic-Spanish-English). *In: Proceedings of the 5th Intl. Conf. on Language Resources and Evaluations (LREC)*, 2006.

Sanderson, M. & Zobel, J. 2005. Information retrieval system evaluation: effort, sensitivity, and reliability. *In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005. ACM, 162-169.

Savoy, J. 2002. Report on CLEF-2001 experiments: Effective combined query-translation approach. *In: Evaluation of Cross-Language Information Retrieval Systems*, 2002. Springer, 45-90.

Savoy, J. 2003. Report on CLEF 2002 experiments: Combining multiple sources of evidence. *Advances in Cross-Language Information Retrieval*, 66-90.

Savoy, J. 2007. Why do successful search systems fail for some topics. *In: Symposium on Applied Computing: Proceedings of the 2007 ACM symposium on Applied computing*, 2007. 872-877.

Sharoff, S. 2006. Creating general-purpose corpora using automated search engine queries.

Sheridan, P., Ballerini, J. P. & Schäuble, P. 1998. Building a large multilingual test collection from comparable news documents. *Cross-Language Information Retrieval*. Kluwer Academic Publication, 137-150.

Si, L., Callan, J., Cetintas, S. & Yuan, H. 2008. An effective and efficient results merging strategy for multilingual information retrieval in federated search environments. *Information Retrieval*, 11, 1-24.

Sigurbjörnsson, B., Kamps, J. & de rijke, M. 2005. Blueprint of a cross-lingual web retrieval collection. *In: Information Retrieval Workshop*, 2005. 33.

Sparck jones, K., Walker, S. & Robertson, S. E. 2000. A probabilistic model of information retrieval: Development and comparative experiments: Part 1. *Information processing & management*, 36, 779-808.

Tayli, M. & Al-salamah, A. I. 1990. Building bilingual microcomputer systems. *Communications of the ACM*, 33, 495-504.

Tong, M. J. K. S. T. 1991. Text Analysis in Computer-assisted Language Learning.

Tony, M. E., Xioa, R. & Tono, Y. 2006. Corpus-Based Language Studies an Advanced Resource Book. London and New York: Routledge.

Ture, F., Lin, J. & Oard, D. W. 2012. Looking inside the box: context-sensitive translation for cross-language information retrieval. *In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012. ACM, 1105-1106.

Van rijsbergen, C. & Sparck jones, K. 1975. Report on the need for and pro-vision of an "ideal" information retrieval collection.

Vergyri, D. & KirchhofF, K. 2004. Automatic diacritization of Arabic for acoustic modeling in speech recognition. *In: COLING 2004 Workshop on Computational Approaches to Arabic Script-based Languages*, 2004. 66-73.

Voorhees, E. M., Gupta, N. K. & Johnson-laird, B. 1995. Learning collection fusion strategies. *In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 1995. ACM, 172-179.

Voorhees, E. M. & Harman, D. 2000. Overview of the sixth text retrieval conference (TREC-6). *Information processing & management*, 36, 3-35.

Voorhees, E. M. & Harman, D. 2001. Overview of TREC 2001. In: Proceedings of TREC, 2001. 1-15.

Wang, J. & Oard, D. W. 2006. Combining bidirectional translation and synonymy for cross-language information retrieval. *In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006. ACM, 202-209.

World's Muslim Population. 2013. World's Muslim Population [Online]. Available: http://islam.about.com/od/muslimcountries/a/population.htm [Accessed 2013].

Xu, J. & Croft, W. B. 1998. Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems (TOIS)*, 16, 61-81.

Xu, J., Fraser, A. & Weischedel, R. 2001. TREC 2001 cross-lingual retrieval at BBN. *In: TREC 2001*, 2001. 68-78.

Xu, J., Fraser, A. & Weischedel, R. 2002. Empirical studies in strategies for Arabic retrieval. *In: Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 25 th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002. 269-274.

Xu, J. & Weischedel, R. 2005. Empirical studies on the impact of lexical resources on CLIR performance. *Information processing & management*, 41, 475-487.

Yang, Y., Carbonell, J. G., Brown, R. D. & Frederking, R. E. 1998. Translingual information retrieval: learning from bilingual corpora. *Artificial Intelligence*, 103, 323-345.

Zawaydeh, B. & Saadi, Z. 2006. Orthographic Variations in Arabic Corpora. *In: Goverments Users Conference*, 2006.

Zhang, J. & Lin, S. 2007. Multiple language supports in search engines. *Online Information Review*, 31, 516-532.

Zhang, Y. & Vines, P. 2004. Using the web for automated translation extraction in cross-language information retrieval. *In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004. ACM, 162-169.

Zhang, Y., Vines, P. & Zobel, J. 2005. Chinese OOV Translation and Post-translation Query Expansion in Chinese-English cross-lingual information retrieval. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4, 57-77.

Zhou, D., Truran, M., Brailsford, T. & Ashman, H. 2008. A Hybrid Technique for English-Chinese Cross-language Information Retrieval. *ACM Transactions on Asian Language Information Processing (TALIP)*, 7, 5.

Zobel, J. 1998. How reliable are the results of large-scale information retrieval experiments? *In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998. ACM, 307-314.

Appendices

Appendix

A

A Mixed Chinese-English query submitted to Google

Appendix

B

Queries Used in the Experiments

ID.	Query	Approximate Meaning
DLIB01	مفهوم الـ deadlock	Concept of deadlock
DLIB02	ماذا نعني بالـ (SSL) Secure Socket Layer	What is meant by Secure Socket Layer (SSL)
DLIB03	الفرق بين ال interpreter و ال assembler	Difference between interpreter and assembler
DLIB04	شرح polymorphism في الجافا	Explain polymorphism in Java
DLIB05	مثال في Entity Relationship Model	Entity and Relationship Model, example
DLIB06	تقنيات Data Mining	Data mining techniques
DLIB07	تمارين Synchronized Methods جافا	Tutorials on synchronized methods in Java
DLIB08	إنشاء table في Oracle	Create table in Oracle
DLIB09	حذف عقدة binary tree كود	Delete node in binary tree, code
DLIB10	رسم تقريبي three tier architecture	Three tier architecture, illustrative figure
DLIB11	مفهوم الـ concurrency control database	Concept behind concurrency control in database
DLIB12	ال overload ضد ال override في سي	Overload versus override in C language
DLIB13	نظام التشغيل Mutual exclusion	Mutual Exclusion in operating system
DLIB14	Multiple inheritance لغة جافا	Multiple inheritance in Java
DLIB15	Assembly لغة registers	Registers in Assembly language
DLIB16	مسائل محلولة Boyce Cod Normal Form	Boyce Cod Normal Form, solved problems
DLIB17	مقارنة بين circuit switching and packet switching	Compare between circuit switching and packet switching
DLIB18	عبارة If في لغة Python	If syntax in Python
DLIB19	تمارين double linked list	Tutorials on double linked list
DLIB20	مفهوم firewall	Concept behind firewall
DLIB21	مثال casting Java script	Casting in Java script, example

* see the next page

ID.	Query	Approximate Meaning
DLIB22	مقدمة Software Engineering	Introduction to software engineering
DLIB23	شكل ال MAC frame	MAC frame, diagram
DLIB24	تطبيقات expert systems النظم الخبيرة	Expert Systems, Applications
DLIB25	شرح clustering index	Explain clustering index
DLIB26	مسائل محلولة relational algebra	Solved problems, relational algebra
DLIB27	مصطلح Paging في نظام التشغيل	Paging in operating system
DLIB28	التشفير باستخدام hashing function	Encryption using Hashing function
DLIB29	تنفيذ برنامج في Android	Execute programs under Android
DLIB30	Oracle performance tuning مثال تطبيقي	Oracle performance tuning, working example
DLIB31	التحكم في الازدحام switched data network	Congestion control in switched data network
DLIB32	مشكلة stable marriage الخوارزميات	Stable marriage problem in algorithms
DLIB33	أمثلة ternary relationship	Ternary relationship, examples
DLIB34	تطبيق ال stack recursion	Implementing stack using recursion
DLIB35	التطبيع normalization في قواعد البيانات	Normalization in database
DLIB36	استخدامات multimedia database	Usage of multimedia database
DLIB37	دعم اللغة العربية content management system	Arabic support, content management system
DLIB38	شرح quick sort	Explain quick sort
DLIB39	أمثلة inner-join and outer-join	Examples, inner-join and outer-join
DLIB40	خوارزمية divide and conquer	Divide and conquer algorithm
DLIB41	التشفير بإستخدام Data Encryption standard DES	Encryption using Data Encryption Standard DES
DLIB42	عيوب ال optical fibers	Drawbacks of optical fibers
DLIB43	عروض تقد يمية cloud computing	Presentations on cloud computing
DLIB44	خصائص الـ object relational database	Characteristics of object relational database
DLIB45	ثغرات الهجوم SQL Injection	Attack vulnerability, SQL injection
DLIB46	الفرق بين meta-data وال catalog	Difference between meta-data and catalog
DLIB47	خوارزميات Warshall and Floyd	Warshall and Floyd algorithms