A Novel Arabic Lemmatization Algorithm

Eiman Al-Shammari
George Mason University
Kuwait University
4400 University Drive, MS4A5
Fairfax, VA 22030
Eiman.tamah@gmail.com

Jessica Lin
George Mason University
4400 University Drive, MS4A5
Fairfax, VA 22030
jessica@cs.gmu.edu

ABSTRACT

Tokenization is a fundamental step in processing textual data preceding the tasks of information retrieval, text mining, and natural language processing. Tokenization is a language-dependent approach, including normalization, stop words removal, lemmatization and stemming.

Both stemming and lemmatization share a common goal of reducing a word to its base. However, lemmatization is more robust than stemming as it often involves usage of vocabulary and morphological analysis, as opposed to simply removing the suffix of the word. In this work, we introduce a novel lemmatization algorithm for the Arabic Language.

The new lemmatizer proposed here is a part of a comprehensive Arabic tokenization system, with a stop words list exceeding 2200 Arabic words. Currently, there are two Arabic leading stemmers: the root-based stemmer and the light stemmer. We hypothesize that lemmatization would be more effective than stemming in mining Arabic text. We investigate the impact of our new lemmatizer on unsupervised data mining techniques in comparison to the leading Arabic stemmers. We conclude that lemmatization is a better word normalization method than stemming for Arabic text.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Indexing methods, Linguistic processing;* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Clustering.*

General Terms

Algorithms, Documentation, Experimentation, Standardization, Languages.

Keywords

Text Mining, Arabic, Stemming, Lemmatization, Tokenization

AND'08, July 24, 2008,Singapore.

1. INTRODUCTION

The Internet is witnessing an explosive growth in the field of information search and retrieval. Unfortunately, due to the language differences, this growth is limited to the language in which it was developed (usually English) and cannot be easily transferred to different linguistic environments. Some languages share similar structures, whereas others are totally different. In such cases, text processing algorithms developed for a specific language cannot be applied to other languages.

Arabic is the sixth most widely spoken language in the world. According to the Global Research's (2004) estimate, there are 10.5 million Arabic speakers with access to the Internet, compared to 287.5 million English speakers. Unfortunately, efforts to improve Arabic information search and retrieval compared to other languages are limited and modest. The barrier to text processing advancements in Arabic is the very complicated morphological structure of the Arabic language.

Stemming is a computational process for reducing words to their root (or stem),[1] and it can be viewed as a recall-enhancing device or a precision-enhancing device. As a result, stemmers are basic elements in query systems, indexing, web search engines and information retrieval systems (IRS).

The current Arabic stemming approaches only focus on the morphological structure. Ignoring Arabic basic rules can cause errors in automatic translation, text clustering, text summarization, and NLP. Currently, there are two Arabic leading stemmers: the root-based stemmer and light stemmer.

The structure of Arabic makes it harder to stem the words to their roots. Common stemming errors that stemmers suffer from include over-stemming , under-stemming, and mis-stemming.

This paper presents a new stemming algorithm that relies on Arabic language morphology and Arabic language syntax. The addition of the syntactical knowledge creates what is known as a lemmatizer in linguistics. The automated addition to the syntactic knowledge reduces both stemming errors and stemming cost.

This paper is organized as follows: In Section 2, we briefly review the Arabic language morphology and discuss previous Arabic language tokenization process. Our methodology is presented in section 3, followed by the proposed lemmatization algorithm in Section 4. In Section 5, we present the evaluation criteria and the experimental results. Finally, we conclude our study and discuss future work in Section 6.

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

Figure 1: Arabic Particles (letters)

2. Background and Related Work

Arabic language is a semantic language with a composite morphology. The words are categorized as particles, nouns, or verbs. There are 29 letters in Arabic, and the words are formed by linking letters of the alphabet. Figure 1 shows a list of Arabic letters. Unlike most Western languages, Arabic script is written from right to left. The letters are connected and do not start with capital letter as in English. Due to the unique characteristics of Arabic language, one particularly challenging task for machines is to recognize and extract proper nouns from Arabic texts.

Furthermore, in English, words are formed by attaching prefixes and suffixes to either or both sides of the root. For example the word *Untouchables* is formed as follows

Un	touch	able	S
Prefix	Root	First Suffix	Second Suffix

In Arabic, additions to the root can be within the root (not only on the word sides) which is called a pattern. This causes a serious issue in stemming Arabic documentation because it is hard to differentiate between root particles and affix particles.

Table 1 displays an example of the Arabic Word = الشارب (drinker) and its stems with the common prefixes and suffixes.

Table 1. Arabic Example

Prefixes + Stem (Root + Pattern) + Suffixes		
Root	شرب	drink
Prefixes	ال	the
Stem	شارب	drinker
Suffixes	ان, ين	dual
Suffixes	ون	plural
Suffixes	ة	feminine
الشاربين الشاربان	the drinkers (dual)	
الشاربين	the drinkers (plural)	
الشارب	the drinker (masculine)	
الشاربة	the drinker (feminine)	

Due to the Arabic morphological structure, Arabic requires a different stemming process from other languages.

Stemming Arabic documentations was done manually prior to TREC (Text Retrieval Conference) and only applied on small corpora. As mentioned, the most common Arabic stemming approaches are the root-based and the light stemmers.

Automatic Arabic stemming proved to be an effective technique for text processing for small collections [2], [3] and large collections [4],[5] of documents. Xu et al. [6] showed that spelling normalization combined with the use of tri-grams and stemming could significantly improve Arabic text processing by 40%.

The two most effective Arabic stemmers are Larkey's light stemmer[4] [5] and Khoja's [7] root-extraction stemmer. In addition, Duwairi [8] , El-Kourd et al.[9] and Mustafa et al.[10] discovered that N-gram stemming technique is not efficient for Arabic Text processing. In summary, Arabic stemming produced promising results in some applications and failed in others.

Over-stemming and under-stemming are the main drawbacks of the root-based stemming and the light stemming algorithms respectively. Over-stemming, under-stemming and mis-stemming are all stemming errors that usually degrade the correctness of stemming algorithms[11].

As stated in[12], mis-stemming is defined as "taking off what looks like an ending, but is really part of the stem," and over-stemming is "taking off a true ending which results in the conflation of words of different meanings".

Arabic stemmers blindly stem all the words and perform poorly especially with compound words, proper nouns and foreign Arabized words. The main cause of this problem is the stemmer's lack of knowledge of the word lexical category (i.e. noun, verb, proposition, etc.)

A possible solution for this problem is to add a lookup dictionary to check the roots. Although this solution seems straightforward and easy, this process is computationally expensive. Al-Fedaghi and Al-Anzi [13] estimated that there are around 10,000 independent roots. Each root word can have prefixes, suffixes, infixes, and regular and irregular tenses.

Another solution is to define a rule to stem words instead of chopping off the letters blindly; this rule is set by the syntactical structure of the word. For example verbs require aggressive stemming and need to be represented by their roots. Nouns on the contrary only require light suffixes and prefixes elimination. This advanced stemming is known as Lemmatization[14].

Lemmatization is a normalization technique[5], generally defined as "the transformation of all inflected word forms contained in a text to their dictionary look-up form" [15]

To our best knowledge there has been no proposed algorithm for Arabic Lemmatization.

In this work, we propose the first Arabic lemmatization algorithm, and we hypothesize that lemmatization will be more efficient in tokenizing Arabic documents than stemming. In addition to the general stemming benefits, lemmatization can overcome the stemming errors and reduce stemming cost by reducing unnecessary stemming.

3. Methodology

Tokenization often performs stop words removal early in the process, although there is currently no standardized list of Arabic Stop Words. The current available list [4] introduces less than 200 words. Table 2,3,4, and 5 show a subset of these words.

We are able to define more than 2,200 stop words and categorize them into useful and useless stop words. Useless stop words are stop words that are used extensively and give no benefits to the subsequent words. Table 3 and 5 are examples of useless stop words. Useful stop words are words that can indicate the syntactical categories of the subsequent words. For example, in an English sentence such as “I went to school yesterday,” it is easy to realize that school is a noun and thus does not require aggressive stemming.

Unfortunately, due to the early removal of the stop words, we lost this valuable information. The same scenario applies to Arabic language too. We believe that the useful stop words can help us identify nouns and verbs and direct us into the appropriate stemming. Our algorithm can also be considered as an advanced stemmer, in which identified nouns and verbs are used to generate global nouns and verbs dictionaries. The benefit of these dictionaries is to find similar nouns in the corpus that were used differently in other sentences. For example in the following paragraph the word *School* is identified as a noun and was recognized as a noun in the following sentence.

I went to school yesterday, I love school.

In table 2, we show sub list of stop words preceding verbs, and table 3 presents some of the stop words preceding nouns. Our stop words list was initially generated by three methods; English stop words translation, identification of common words in arbitrary Arabic documents, and manual search of synonyms to the previously identified stop words.

In the following section we will describe our algorithm in details.

Table 2. Preposition Preceding Verbs

Preposition	English
حيثما	Wherever
كلّما	Whenever
إذا	If
عندما	When (not for question)

Table 3. Arabic circumstantial nouns indicating time and place

Preposition	English Equivalence
بعد	After
على	Over
فوق	Above, up
إلَى	until ,near, towards ,to
أمامَ	in front of:
باتجاه	On the direction of
بجانب	Aside, next to, beside
تحت	Below, beneath, down
حتى	Till (time and location)
خارج	Outside of
خلال	Through, during,
عبر	Through
عن	From, about
في	In (time, location, duration)
قبلَ	Before
قريب	Near
منذ	since
وراء	Behind , Beyond
بين	Between

Table 4 . Arabic Independent Pronouns

Word	English Equivalence
نحن	Us
أنا	I am
أنتم /أنت	You (feminine/Masculine) Plural
أنتن	You (feminine/Masculine) Singular
هو /هي	She/he
هم/هن	Them (feminine/Masculine)
هما	Them (dual)

Table 5. Arabic Demonstrative prepositions

Preposition	English Equivalence
هذا	This : used for masculine
هذه	This : used for feminine
ذلك	That : used for masculine
تلك	This : used for feminine
أولئك	These
هؤلاء	Those

4. Arabic Lemmatization Algorithm

As shown in Figure 2, our novel algorithm consists of different phases. During the first phase, useless stop words are removed to reduce the size of the corpus. Next, we identify nouns by either locating noun preceding stop words or words starting by definite articles. These nouns are lightly stemmed by removing suffixes and prefixes and then added to the global nouns dictionary[16]. At this level, these words are flagged as nouns as a preparation to the stemming phase. In parallel to that process we find verbs by locating verbs preceding stop words. Similar to the nouns, the verbs are added to the global verb dictionary and tagged as verbs. In Arabic, we cannot have two consecutive verbs, thus any word following a verb is either a stop word or a noun. If the word is not a stop word then the word is added to the noun dictionary and flagged as a noun.

Before we direct a word to the appropriate stemming by the word flag, all the stop words are removed since they offer no further advantage. Other words that do not belong in any category will be treated as nouns and stemmed lightly.

Table 6 below summarizes the algorithm.

Table 6. Arabic Lemmatization Algorithm

```
Input: Arabic document
Output: Stemmed document.
        Noun Dictionary.
        Verbs Dictionary.

    V: Verb dictionary (one dimensional array sorted alphabetically[1])
    N: Noun dictionary (one dimensional array sorted alphabetically)
    NSW: Array of stop words proceeding nouns
    VSW: Array of stop words proceeding verbs
    SW: Array of stop words (including both NSW and VSW)

Phase Zero: Remove useless stop words.
```

---

[1] For fast lookup, these dictionaries can be implemented using hash tables

Phase One: Simple Noun identification

Locate words attached to definite articles, and preceded by NSW and flag them as Nouns

Phase Two: Suffix and Prefix removal

Apply suffix and prefix approach to the entire document. Longest suffixes and prefixes are removed first.

Phase Three: Noun Dictionary Generation

Add the identified processed words to N.

Phase Four: Verbs Identification

Verbs are proceeded by VSW

Phase Five: Verb Dictionary Generation

Add the identified processed words to V

Phase Six : Find all noun tokens

Phase Seven: Stop Word Removal

Remove useful and useless stop words

Phase Eight: Root Extraction for Verbs

Roots are extracted by comparing Verbs to Arabic Root patterns, words (tokens) with missing tags are considered nouns and lightly stemmed.

5. Evaluation and Experiments

We compare our Arabic Lemmatization algorithm to the leading Arabic root-based stemmer presented by Khoja. We observe the stemmer effects on improving the document clustering performance in comparison to the Khoja stemmer.

5.1 Experiment setting

5.1.1 Data Description

In our experiments, we use modern, unedited, and unmarked Arabic text, a sample of approximately 7000 documents from various Arabic online resources to construct three datasets. The first dataset contains economical articles drawn randomly from Al-Watan (2008), a newspaper from Kuwait. The second dataset is extracted from Arabic medical websites. This dataset contains two subsets: Kidney failure related articles and physiology related documents. The last dataset contains randomly selected documents retrieved from the ACE2004 (ACE 2004 Multilingual Training Corpus, Linguistic Data Consortium)

5.1.2 Clustering

Cluster analysis [17] is the process of dividing objects into groups of similar objects according to a distance measure. Clustering is applied in many fields including text mining and machine learning. K-means is a widely used partitional clustering method with a linear time complexity[18] .

To design our experiments we chose to study our algorithm's effect on improving the performance of the K-means clustering in contrast to the Khoja stemmer's effect. We used the TFIDF weighted function and created three different experiments sets.

The first and the second experiments are to cluster documents belonging to two different groups. The last experiment is to study the effect on clustering three different groups. More details are included in the results subsection.

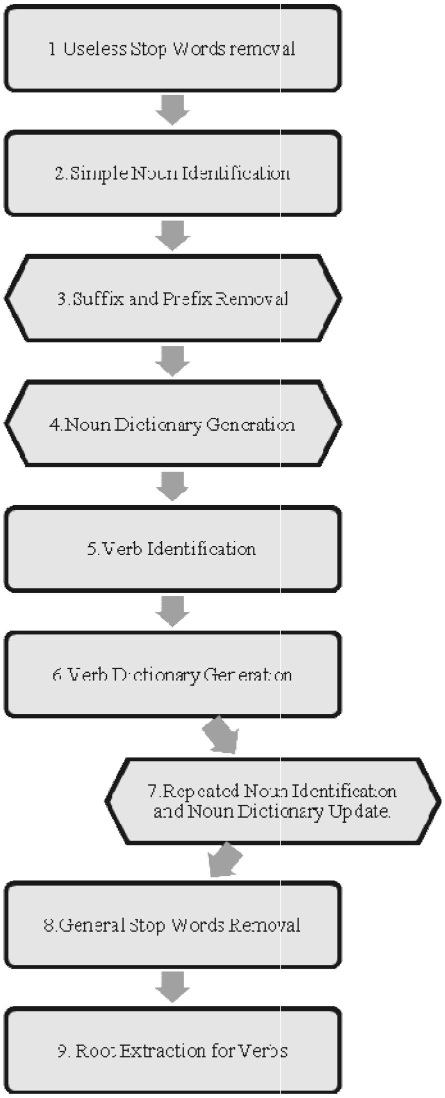
1 Useless Stop Words removal
2. Simple Noun Identification
3. Suffix and Prefix Removal
4. Noun Dictionary Generation
5. Verb Identification
6 Verb Dictionary Generation
7. Repeated Noun Identification and Noun Dictionary Update
8. General Stop Words Removal
9. Root Extraction for Verbs

Figure 2. The Lemmatization Algorithm Simplified

5.1.3 Performance Measure

To measure the quality of clusters we use the Cluster Purity [19]which is the percentage of documents correctly labeled. The overall purity is the weighted sum of individual clusters purities. The algorithm that will lead to a better clustering is the algorithm generating a higher overall purity.

5.2 Results

We perform three different experiments designed to compare the Lemmatization and the Stemming effects on improving the purity of K-mean clustering. On the first experiment our goal is to study the effect on highly relevant documents, thus we chose the two medical subsets. Both Khoja stemmer and our lemmatizer achieved the same clustering purity. In the second experiment, we choose two contrasting datasets,: Medical and News, The clustering purity for the lemmatized documents was 10% higher than the clustering purity for the stemmed documents.

This result was expected since the Khoja stemmer tends to over stem words, which leads to creating similarities between unrelated documents containing same roots for different words.

Finally, for our last experiment, we choose three contrasting datasets: News, Economics , and Medical.

Applying K-means clustering on the three datasets leads to an overall cluster purity of 70.8% for the lemmatized documents and 58% for the stemmed document.

We examine the characteristics of each cluster and notice that medical and economical documents are mis-clustered due to the existence of many similar words like high and low (i.e. High temperatures and high stock prices).

6. Conclusion and Future Work

In this paper we introduce the first Arabic Lemmatization Algorithm and compare its performance with the Khoja stemming Algorithm for clustering applications.

Additionally we introduce a new framework to normalize Arabic documents by overcoming the limitations of previous approaches, caused by the early removal of stop words. We show that Arabic neglected stop words can be highly important and can provide a significant improvement to processing Arabic documents. The approach also can reduce English documentation stemming errors due to the prior knowledge of nouns and proper nouns[20].

Stemming error is a subjective measure that does not get much attention in comparing Arabic stemmers. We perform initial experiments on the output of our lemmatizer and on the Khoja stemmer's output and find out that the Khoja stemmer have a high over-stemming error rate.

Our experiments show a promising future for our lemmatizer, which encourages us to apply further research on clustering larger number of documents. Currently we are working on studying lemmatization on Arabic document classification. Also, we would like to study the lemmatizer effect on precision/recall, as well as stemming cost, in comparison to the Khoja stemmer and other Arabic stemming Algorithms.

7. REFERENCES

[1] W.B. Frakes, "Stemming algorithms," 1992.

[2] I.A. Al-Kharashi, "Micro-AIRS: A microcomputer-based Arabic information retrieval system comparing words, stems, and roots as index terms," 1991.

[3] I.A. Al-Kharashi and M.W. Evens, "Comparing Words, Stems, and Roots as Index Terms in an Arabic Information Retrieval System.," *Journal of the American Society for Information Science*, vol. 45, 1994, pp. 548-60.

[4] L.S. Larkey and M.E. Connell, "Arabic Information Retrieval at UMass in TREC-10," *Proceedings of the Tenth Text REtrieval Conference (TREC-10)", EM Voorhees and DK Harman ed*, 2001, pp. 562-570.

[5] L.S. Larkey, L. Ballesteros, and M.E. Connell, "Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis," Tampere, Finland: ACM, 2002, pp. 275-282.

[6] J. Xu, A. Fraser, and R. Weischedel, "Empirical studies in strategies for Arabic retrieval," *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 269-274.

[7] S. Khoja and R. Garside, "Stemming Arabic Text," *Lancaster, UK, Computing Department, Lancaster University*, 1999.

[8] R. Duwairi, "A Distance-based Classifier for Arabic Text Categorization," *Proceedings of the 2005 International Conference on Data Mining, Las Vegas USA*, 2005.

[9] M. EL KOURDI, A. BENSAID, and T. RACHIDI, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," *COLING 2004*.

[10] S.H. Mustafa and Q.A. Al-Radaideh, "Using N-grams for Arabic text searching," *Journal of the American Society for Information Science and Technology*, vol. 55, 2004, pp. 1002-1007.

[11] R.A. Baeza-Yates, "Text-Retrieval: Theory and Practice," North-Holland Publishing Co., 1992, pp. 465-476.

[12] "Snowball: A language for stemming algorithms"; http://snowball.tartarus.org/texts/introduction.html.

[13] S.S. Al-Fedaghi and F. Al-Anzi, "A New Algorithm to Generate Arabic Root-Pattern Forms," *Proceedings of the 11th National Computer Conference and Exhibition*, 1989, pp. 391–400.

[14] T. Korenius et al., "Stemming and lemmatization in the clustering of finnish text documents," Washington, D.C., USA: ACM, 2004, pp. 625-633.

[15] M. BOOT, “Homography and Lemmatization in Dutch Texts,” *ALLC Bulletin*, vol. 8, 1980, pp. 175-189.

[16] Eiman Al-Shammari and J. Lin, “Automated Corpora Creation Using A novel Arabic Stemming Algorithm,” *The 2008 International Symposium on Using Corpora in Contrastive and Translation Studies (UCCTS)*, Hangzhou, China: 2008.

[17] A.K. Jain and R.C. Dubes, *Algorithms for clustering data*, 1988.

[18] M. Steinbach, G. Karypis, and V. Kumar, “A comparison of document clustering techniques,” *KDD Workshop on Text Mining*, vol. 34, 2000, p. 35.

[19] Y. Zhao and G. Karypis, “Criterion Functions for Document Clustering,” *Experiments and Analysis University of Minnesota, Department of Computer Science/Army HPC Research Center*.

[20] E. Al-Shammari, “Towards an Error Free Stemming,” *IADIS European Conference on Data Mining (ECDM 2008)*, Amsterdam, The Netherlands: 2008.