A Novel Context-based Technique for Web Information Retrieval

John Zakos
School of Information Technology
Griffith University
Queensland 4215 Australia
j.zakos@griffith.edu.au

Brijesh Verma
School of Information Technology
Central Queensland University
Queensland 4702 Australia
b.verma@cqu.edu.au

Abstract: In this paper we present *context matching*, a novel context-based technique for the ad-hoc retrieval of web documents. The aim of the technique is to dynamically generate a measure of document term significance during retrieval that can be used as a substitute or co-contributor of the term frequency measure. Unlike term frequency, which relies on a term occurring multiple times in a document to be considered significant, context matching is based on the notion that if a term in a given document occurs in that document in the context of the query, then that term is deemed to be significant. Context matching has the ability to potentially determine a term to be significant even if it occurs only once in a document. Vice versa, it also has the ability to determine a term to be insignificant, even if occurs frequently within a document. We show how expanded terms generated by a typical query expansion technique can be used effectively as query context for context matching. The technique is ideally suited to the nature of web information retrieval and we show how context matching significantly improves retrieval accuracy through experimental results on TREC web benchmark data.

Keywords: web information retrieval, context-based, term weighting, query expansion.

Running Head: Context-based Web Information Retrieval.

1 Introduction

Traditionally, a term in a given document is considered to be significant if it occurs multiple times within that document. In fact, the more times a term occurs in a document then the more significant it is deemed to be. This observation, commonly referred to as term frequency (TF), was made by Luhn [18], where he noticed that authors of documents typically emphasize a subject or concept by repeatedly using the same word/s.

Since then, most information retrieval approaches [4] have adopted TF (or variations of it) as the standard way of indicating how significant or relevant a term is to a given document. In particular, it is normally combined with inverse document frequency (IDF) to form the TFIDF measure [24]. This has been popularized in the vector space model [25].

Even with the emergence of web information retrieval, TF still remains as the salient measure of term significance within a document. There are several examples content-based web information retrieval systems [17,8,23,3] that base their determination of term significance through TF. In the OKAPI weighting scheme [27,23], the TF component has a number of parameters that can be tuned to obtain optimum performance [11].

But as is the case for many potentially relevant documents, TF is not always the best or most useful indicator of term significance or relevancy. Quite often, there are relevant documents that contain only a single or a few occurrences of a particular term. Consequently, through TF these terms will rarely be considered significant, and thus never contribute greatly to the rank score of the potentially relevant document they appear within. This is especially the case when infrequently occurring terms appear in large documents containing hundreds or even thousands of terms.

A technique called *context matching* (CM) is presented in this paper. It is a technique that generates term confidence in a fundamentally different way to TF. CM does not rely on the number of times a particular term appears in a document to determine whether it is significant or not. With CM, a term appearing infrequently in within a large document can potentially be given a high confidence. Vice versa, a term appearing frequently within a document can potentially be assigned a low confidence. This is a significant characteristic of CM that makes it different to TF.

Another fundamental difference in the technique lies in the way expanded terms generated by query expansion are used. Instead of adding expanded terms to the original query, CM uses expanded terms from query expansion to form the query context. This context is then used as a basis for matching contexts with terms in documents and ultimately calculating term significance. Traditionally, query expansion, whether based on global or locally retrieved documents [29] or a knowledgebase [26], is a technique that has been shown to improve retrieval accuracy by adding expanded terms to the original query. It aims to overcome the keyword mismatch problem and contribute to boosting documents that would otherwise be lowly ranked or possibly never retrieved. In a recently proposed technique, Yu et al [30] propose a visual based page segmentation algorithm to assist query expansion in selection of expanded terms for web information retrieval.

In other query-centric approaches to retrieval [15,21], queries can be classified to aid in the choice of retrieval strategy. Kang et al [15] classify queries as either pertaining to topic relevance, homepage finding or service task and use this classification as a basis of dynamically combining multiple evidences in different ways to improve retrieval. Plachouris et al [20] use WordNet in a concept-based probabilistic approach to information retrieval where queries are biased according to their calculated scope. In their work, scope is an indication of generality or specificity of a query and is used as a factor of uncertainty in Dempster-Shafer's theory of evidence. In another approach [21], query scope is determined through statistical measures derived from a set on initially retrieved documents and used to as a basis decide the type of retrieval strategy (i.e. content-only vs. content and hyperlink).

The use of context in information retrieval is not a new idea. Jing et al [14] use context as a basis of measuring the semantic distances between words. During indexing, the context of terms in documents is generated and stored in vector form. During retrieval, the context of a term in a query is generated and is used to measure the semantic distance between itself and

candidate morphological variants in documents. Mutual information of terms is used to match related terms during the calculation of context distance.

Billhardt et al [6] propose a context-based vector space model for information retrieval. After the term-document matrix has been constructed, it is used is a basis for generating a term context matrix where each column is considered a semantic description of a term. This term context matrix is then combined with the document vectors from the term-document matrix to transform it into the final document context vector used for retrieval. They report 28% improvement in retrieval accuracy when using the context-based approach.

The WEBSOM [13] system is an example of another way in which context has been used for information retrieval. It uses a two level Kohonen's self-organizing map approach to group words and documents of contextual similarity. Context in WEBSOM is limited to the terms that occur direct either sides of the term in question.

IntelliZap [9] is a context-based web search engine that requires the user to select a key word in the context of some text. The approach makes effective use of the contextual information in the immediate vicinity of the keywords selected, so that retrieval precision can be improved. Inquirus [10,17] is another web search engine that uses contextual information to improve search results. A user must specify some contextual information, considered as preferences, pertaining to the query. This context (preferences) provides a high-level description of the users information need and ultimately control the search strategy used by the system.

Hyperlink information can be a very valuable source of evidence for web information retrieval and it is either based on a set of retrieved documents during retrieval or on a global analysis of the entire document collection during indexing [12]. Kleinberg [16] illustrates how hyperlink information in web pages can be used for web search when using a set of retrieved documents. Kleinberg's algorithms are based on the notion that if page *p* points to page *q* then *p* has some measure of conferred authority on *q*. Thus, the more pages that point to *q*, the more authoritative *q* should be as an authority of information on the topic it represents.

Bharat et al [5] propose algorithms based on Kleinberg's work. They identify some inherent problems with using hub and authority calculations from neighbourhood graphs such as mutually reinforcing relationships between hosts, automatically generated links and non-relevant nodes. An approach that also uses the characteristics of link information from a set of retrieved documents for topic distillation is presented by Amitay et al [2].

PageRank, as proposed by Brin et al [7], is hyperlink-based retrieval algorithm that calculates document scores by considering the entire hyperlink connected graph represented by all the links in the entire document collection. It uses link information to model user behaviour by calculating the probability that a user will eventually visit a certain page. This probability or PageRank of a page is used to prioritise its ranking during retrieval.

The remainder of paper is organized into 4 further sections. The next section describes the proposed technique. Section 3 presents the experimental setup. Section 4 presents an analysis and discussion of results and in section 5 we conclude and outline future direction of research.

2 Proposed Technique

The context of both terms in documents and terms in queries is fundamental to CM. Its aim is to dynamically determine the significance of a term in a document using the query context of the submitted query. The technique is based on the notion that if a term occurs in a document in the same context as the query, then that term is deemed to be significant to that document. The result of the technique is the generation of a *context matching confidence* (CMC) for a term, which is a measure of term significance that can potentially be used as a substitute or co-contributor of TF as a term confidence measure.

Given a query *Q* and a document collection *DC*, the technique can be used in a retrieval process in the following way:

Step 1 Generate query context *QC* for *Q*
  Step 1.1 Retrieve initial documents *ID*
    Step 1.2 Perform query expansion on *ID* to obtain expanded terms *QR*
    Step 1.3 Form query context *QC* from *Q* and *QR*
Step 2 Retrieve documents
  Step 2.1 For every query term *q* in *Q*
    Step 2.1.1 For every document *D* in *DC* containing *q*
      Step 2.1.1.1 Match *QC* with the term context of term *q* in *D* to calculate context matching confidence *CMC*
      Step 2.1.1.2 Calculate term frequency-based measure *TF* of term *q* in *D*
      Step 2.1.1.3 Combine *CMC* with *TF* to give term confidence *TC* for *q* in *D*
      Step 2.1.1.4 Add *TC* to rank score of *D*

There are a few important aspects of CM that make it unique and different from existing techniques. Firstly, unlike traditional query expansion that adds expanded terms to the original query, CM interprets expanded terms to be a set of terms representing the context of the query. Secondly, unlike TF that relies on a term to occur many times within a document to be considered significant or infrequently to be considered insignificant, CM generates CMC based on the notion that a term in a document is significant only if it occurs in the context of the query. This makes it independent of frequency. Through CM, A term appearing frequently within a document can potentially be assigned a low confidence. Thirdly, CM is dynamic, calculating term significance at retrieval time rather than during indexing.

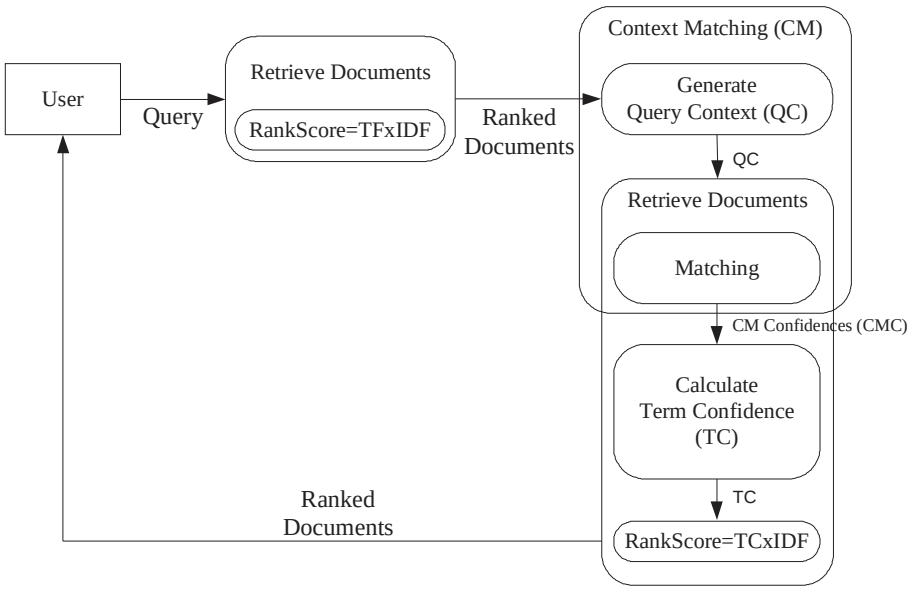

Figure 1. Overview of context matching as part of the retrieval process

The technique is described in detail over the following sections.

2.1 Query Context

The context of a query consists of two sub-contexts, each of which are a set of terms with corresponding relatedness values:

1. Set original query terms *Q*, and
2. Set of related terms *QR*.

These two sets of terms are sub-contexts and together they form the query context $QC = \{Q, QR\}$. Each term in each set has a relatedness value R in the range [0,1] that indicates how related that term is to the original query Q. A value of 1 indicates maximum relatedness where a value of zero indicates that the term is not related. By default all terms have a relatedness value set to 1, unless assigned otherwise.

To determine the set of related terms QR for the query Q, the technique relies on the use of query expansion using local feedback. Typically, an initial run is executed to obtain an initial list of ranked documents and the terms of the top n documents are assumed to be relevant. These n documents are then interpreted and the best m terms are extracted to make up the set QR. We employ the use of

$$TSV_t = w.r \quad (1)$$

to rank candidate terms, where w is a weight (typically IDF) indicating the significance of term t, and r is the number of assumed relevant documents t appears in. This same method to select expanded terms has been used successfully for traditional query expansion [21,22]. Terms are ranked using TSV and the top m are chosen to form QR. Once QR has been determined, it is used as part of the query context QC that can now be used for matching.

2.2 Matching

The aim of matching is, using the query context, to determine the confidence that a term in a document is significant to that document. If a query term occurs in a document and it occurs in the context of the query, then it is considered to be important and given a high confidence of significance. Given a term q and a set of terms that constitute a context C (i.e. Q or QR), then the *contextual importance* (CI) of the occurrence of q in document D can be calculated

$$CI_{q,C,D} = \frac{\sum_{c \in C, c \neq q} \text{Dist}(CD_{q,c,D}) \times R_c}{\sum_{c \in C, c \neq q} R_c} \quad (2)$$

where

c is a term in the context C,
$CD_{q,c,D}$ is the minimum distance between all of the occurrences of q and c in D,
R_c is the relatedness of c to the original query Q (see Section 2.1),
Dist($CD_{q,c,D}$) is a function of distance importance that returns a value in the range [1,0].
The smaller $CD_{q,c,D}$ is the closer Dist($CD_{q,c,D}$) will be to 1. This function can be of type: Gaussian, hard limiter or linear (see below).

Equation 2 is in effect performing matching of the contexts during retrieval. The best match is when terms in C occur directly next to occurrences of q in D. (During indexing, the position of the term in the document is recorded and stored in the index.)

For each query term, the technique separately calculates contextual importance using both the original query Q and related terms QR as contexts. The final measure is the *context matching confidence* (CMC), which is a combination of the CI from both sub-contexts Q and QR. Given a query term q in the query Q, its CMC is calculated by

$$CMC_{q,D} = (CI_{q,Q,D} \times w1) + (CI_{q,QR,D} \times (1 - w1)) \quad (3)$$

where $w1$ is a weighting factor that is set to 0.5 by default.

The resultant CMC is a value in the range [0,1] where a value close to 1 indicates a high confidence that the term q occurring in document D is a significant term and important indicator of relevance for D given Q. A value close to zero indicates insignificance and a low confidence of relevancy. The more related terms (terms in the context) that occur at a closer distance to the occurrence of the query term in the document, the higher the resultant

confidence. On the other hand, the less related terms that occur a further distance from the occurrence of the query term, the lower confidence.

Unlike TF, that calculates term significance by counting the number of times a term occurs within a document, CM relies on the context of the query and the context of term in the document to determine importance. Consequently, it has the significant advantage of potentially giving high confidence to terms that occur infrequently within documents. For example, consider a document *D* where term *q* occurs only once and each term in *QC* occurs only once. If *q* is close to the occurrences of terms in *QC*, then the resultant $CMC_{q,D}$ will be high. This is further exemplified through the use of closest distance $CD_{t,c}$ in Equation 2. Even if both terms *t* and *c* appear only once in a document, if they are close to each other then their relationship of proximity will contribute towards a high CMC score. On the other hand a term that appears frequently within a document and not in the proximity of terms in *QC*, will be given a low $CMC_{q,D}$.

When calculating proximity through distance, a window size is used to define the context area of an occurrence of a term in a document. The area up to *d* (distance) positions to the left and right of an occurrence of term in a document is the context of that term for that occurrence. This is captured through the various distance functions.

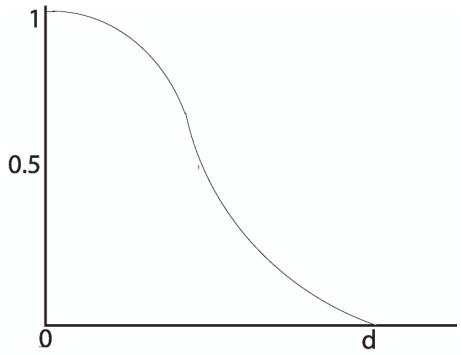

Figure 2. Gaussian function

The Gaussian function as a measure of distance importance takes the form shown in Figure 2. We use a simplified Gaussian formula

$$\text{Dist}(CD) = e^{-\frac{(CD-1)^2}{2\sigma^2}} \quad (4)$$

where σ is Gaussians interpretation of standard deviation and is set to *d*/3, where *d* is the distance outer bounds. When *CD*-1 nears *d* the function returns close to zero. When *CD*-1 nears 0, the function returns 1 or close to 1. This also applies for the linear function of distance that follows the form

$$\text{Dist}(CD) = \frac{d-(CD-1)}{d} \quad (5)$$

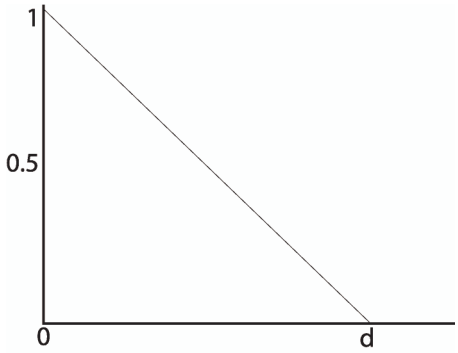

Figure 3. Linear function

The hard limiter function returns only 0 or 1 depending on whether *CD* is greater than *d* or not

$$\begin{array}{ll} \text{Dist}(CD) = 1 & CD - 1 <= d \\ 0 & CD - 1 > d \end{array} \quad (6)$$

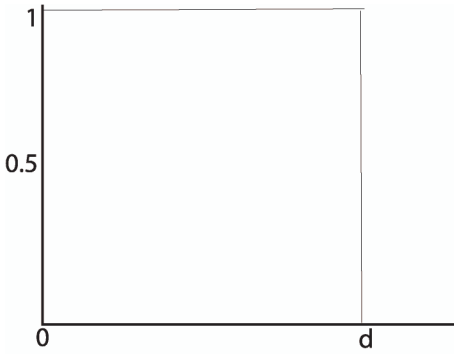

Figure 4. Hard limiter function

The matching technique heavily favours terms occurring very close to each other. The further they are apart or the closer to a distance of *d*, the less important the relationship of proximity is. Looking at Equation 2, we can see that the more terms in the context that occur closer to the term, the more instances of smaller *CD* values there will be. This in turn will result in higher Dist(*CD*) values, which in turn contributes to higher CMC values.

2.3 Combining CMC and TF

As mentioned in the introduction, TF has long been used as a reliable indicator of term significance. We chose to incorporate it into CM by combining it with CMC to give a final confidence measure of a term in a document. Given that matching has been performed and we have a CMC value for a term *q* in a document *D*, the final step of the technique is to combine CMC with TF to give a final term confidence measure. TF is calculated by

$$TF_{q,D} = \frac{\log(count_{q,D} + 1)}{\log(numWords_D + 1)} \quad (7)$$

where $count_{q,D}$ is the number of times *q* occurred in *D*, and *numWords* is the number of terms in document *D*. $TF_{q,D}$ is calculated during indexing. Having both $TF_{q,D}$ and $CMC_{q,D}$, the *term confidence* (TC) of *q* in *D* is calculated by:

$$TC_{q,D} = (TF_{q,D} \times w2) + (CMC_{q,D} \times (1 - w2)) \quad (8)$$

during retrieval, where *w2* is a weighting factor that is set to 0.5 by default.

3 Experimental Setup

Ad-hoc retrieval experiments were run on the TREC benchmark web document collection WT2g, which consists of 247,491 web documents along with 50 queries with corresponding relevance judgements. A standard inverted index was used to index the collection and each node in the index stored a document ID, TF and each position of the term in the document. We use up to 2 bytes (16 bits) for term position. This, 65535 is the largest term position given to a term in a document. (Any terms in documents exceeding position 65535 are applied a threshold and assigned 65535.) Given a query *Q*, the retrieval function used to calculate the score for document *D* is

$$score_{D,Q} = \sum_{q \in Q} TC_{q,D} \times IDF_q \quad (9)$$

where q is a term in the query, $TC_{q,D}$ is the term confidence of term q in document D and IDF_q is the inverse document frequency of q

$$IDF_q = \log_2 \frac{N}{n_q} + 1 \quad (10)$$

where N is the number of documents in the collection and n_q is the number of documents in which term q occurs. To obtain a list of initial documents from which the query expansion technique could extract related terms for *QR*, standard TFxIDF was used

$$score_{D,Q} = \sum_{q \in Q} TF_{q,D} \times IDF_q \quad (11)$$

This run also acts as the baseline for comparison against all other runs. Our experimental investigations followed the strategy:

1. To firstly determine the ideal parameters for *d*, *m* and function Dist(*CD*). For each experiment *d* was set to 10, 30, 50, 100, 250, 1000 or 65535. *m* was set to either 3, 5, 10 or 20. Dist(*CD*) was set to either Gaussian, linear or hard limiter. We ran experiments with all combinations of these parameters. For all experiments $n = 20$ to retrieved the top ranked documents of the initial for query expansion/context generation.
2. To then determine the ideal value for parameter *n* by experimenting with *n* = 2,5,10,30,50.
3. To determine the ideal values of *w1* and *w2* by testing all combinations of values on the range [0,1] at intervals of 0.1.
4. To investigate the impact of using different techniques to obtain relatedness *R* for context terms. This included setting *R* = 1, *R* = *TSV*, *R* = *IDF*.

For all experiments the top 1000 documents are used for evaluation. Obviously we were interested in observing which parameters yielded the best results, but we were particularly interested in comparing the effectiveness of retrieval of CM against traditional query expansion. Also, comparisons against the baseline performance and previous results would give a good indication of the performance of the technique. The results of the experiments along with a discussion and analysis is presented in the next section.

4 Results and Analysis

4.1 Baseline Run

Table 1 shows the result of the baseline run, which uses Equation 11 for retrieval. We can see that an average precision of 0.2987 was achieved.

Table 1. Result of the baseline run

Avg. Prec.	Prec. @ 20	#Rel. Docs
0.2987	0.346	1775

This is the most basic run with just the original query being used for retrieval in a standard TFxIDF approach.

4.2 Query Context

The generation of related terms as a sub-context of a query is an important step of CM. It is vital to generate a sub-context of related terms that relate well to the theme of the query and ultimately to the context of terms in documents. The following is a list of original queries *Q*, along with the top 10 expanded terms that constitute *QR* that were obtained through query expansion:

Q = parkinsons Disease
QR = { dopamine, neurons, brain, levodopa, alzheimer's, dementia, disorder, patients, dyskinesia, substantia}

Q = tropical storms
QR = { trps, cyclones, hurricanes, hurricane, rain, rainforests, cyclone , amazonia, pacific, typhoons}

In general, the terms in the query contexts *QR* are fairly related to the original query, making for good context. Initially, all terms is both *Q* and *QR* are given default relatedness values of 1, indicating that each term is fully related to the original query.

4.3 Results of Traditional Query Expansion

Table 2 shows the results of runs utilizing traditional query expansion. Here, the expanded terms that constitute *QR* are added to the original query and Equation 11 is again used for retrieval.

Table 2. Results for traditional query expansion

m	Avg. Prec.	%Δ	Prec. @ 20	#Rel. Docs
20	0.2534	-15.16%	0.278	1715
10	0.2639	-11.65%	0.309	1781
5	0.2994	+0.25%	0.334	1818
3	0.2986	-0.05%	0.332	1811

As can be seen, traditional query expansion performs worse than the baseline run when m is 20, 10, 3. This may be due to the fact that the expanded terms are not re-weighted except for being assigned their corresponding IDF. This is consistent with the work of Robertson [22] though, who states that terms selected as expanded terms should simply be added to the original query and this query should then be re-submitted for retrieval. Yu et al [30] choose to re-weight original and expanded terms after expansion and report improved results, but we chose to follow a simple and standard approach to expansion and not perform any re-weighting. The fact that no pruning of expanded terms (i.e. to remove very specific or very general expanded terms) is performed could also be contributing the average performance of these traditional query expansion runs. The focus of this aspect of the research was not to improve or advance query expansion. We simply wanted to implement a standard and baseline type expansion algorithm and keep it as uncomplicated as possible.

4.4 Determining *m*, *d*, Dist(*CD*) and *n*

Table 3. Top 10 results when using expanded terms for *QR* in Context Matching

m	*d*	Dist(*CD*)	Avg. Prec.	%Δ	Prec. @ 20	#Rel. Docs
10	250	Linear	0.4142	+38.68%	0.4170	1864
5	100	Linear	0.4137	+38.49%	0.4200	1864
5	250	Gaussian	0.4136	+38.47%	0.4330	1853
10	100	Linear	0.4135	+38.44%	0.4170	1864
10	250	Gaussian	0.4130	+38.27%	0.4160	1858
5	250	Linear	0.4125	+38.10%	0.4320	1859
5	100	Gaussian	0.4112	+37.66%	0.4200	1869
20	250	Linear	0.4097	+37.15%	0.4170	1861
10	100	Gaussian	0.4095	+37.09%	0.4150	1867
20	100	Linear	0.4092	+37.00%	0.4100	1869

In contrast to traditional query expansion, CM performs remarkably well when using the same expanded terms for QR. We ran 84 experiments, testing all combinations of *m*, *d* and Dist(*CD*). Table 3 shows the top 10 runs, ordered by average precision. All runs use Equation 9 for retrieval. The best run which uses 10 expanded terms, a distance of 250 and a linear distance function, achieves an average precision of 0.4142, 38.68% better than the baseline run. Infact, all CM runs (only of which the top 10 are shown in Table 3) comfortably outperformed the baseline run and the traditional query expansion runs. All of the top 10 results shown in Table 3 utilized 5 or more terms for context at a distance of 250 or 100. Also, all of them utilized linear or Gaussian functions for distance. This confirms that observation that terms appearing in a close context to each other are a good indicator of significance. The linear and Gaussian functions capture this by rewarding smaller distances with a value closer to 1, where as the hard limiter distance function ignores with its constant return value for all values smaller than *d*. This is reflected in the results.

Table 4 shows the 10 worse performing runs. The worse performing CM run was when *m* = 3, *d* = 65535 using the Gaussian distance function. This is not surprising at all as 3 terms do not provide much contextual information and 65535 positions is a large context area to be considering during matching. This is effectively the same as ignoring distance and just observing whether the terms co-occur in the same document. This run though still achieved an average precision of 0.3428, still 14.75% better than the baseline run.

Table 4. Bottom 10 results when using expanded terms for QR in Context Matching

m	*d*	Dist(*CD*)	Avg. Prec.	%Δ	Prec. @ 20	#Rel. Docs
20	65535	Linear	0.3536	+18.38%	0.3760	1854
5	65535	Gaussian	0.3531	+18.23%	0.3770	1839
10	65535	Gaussian	0.3527	+18.09%	0.3850	1864
20	65535	Gaussian	0.3517	+17.74%	0.3760	1851
10	65535	Hard Limiter	0.3514	+17.64%	0.3850	1864
20	65535	Hard Limiter	0.3499	+17.15%	0.3720	1850
5	65535	Hard Limiter	0.3494	+16.98%	0.3770	1839
3	65535	Linear	0.3447	+15.40%	0.3760	1850
3	65535	Gaussian	0.3428	+14.75%	0.3740	1849

The fact that CM perform considerably better than traditional query expansion is quite significant. Traditional query expansion techniques that add expanded terms to the original query have been long accepted as the most effective way of dealing with expanded terms. CM presents a novel and effective way of using expanded terms to improve retrieval.

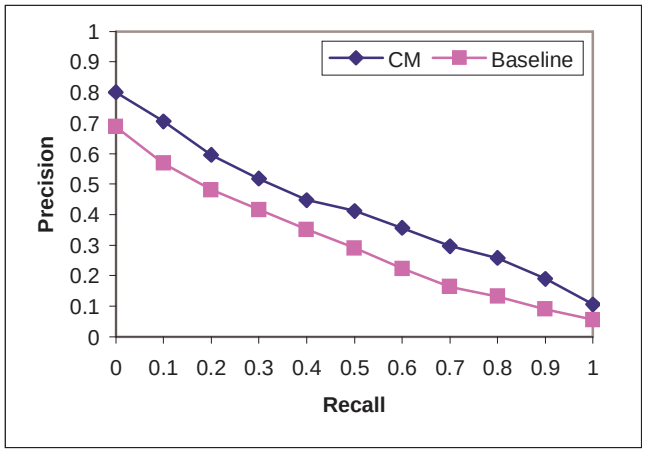

Figure 5. Interpolated recall-precision graph: best CM run vs. baseline

Figure 5 shows that across all levels of recall, CM comfortably outperforms the baseline run. Figure 6 shows the overall performance of each of the parameters. The distance parameter *d*, is the most significant parameter as there is a large variations of average precision across its different values. It seems that a distance *d* of 100 or 250 or somewhere in between is an ideal setting for *d*. Gaussian and linear distance functions are confirmed as the best performing functions. As for the number of terms *m*, around 5-10 is to be most effective.

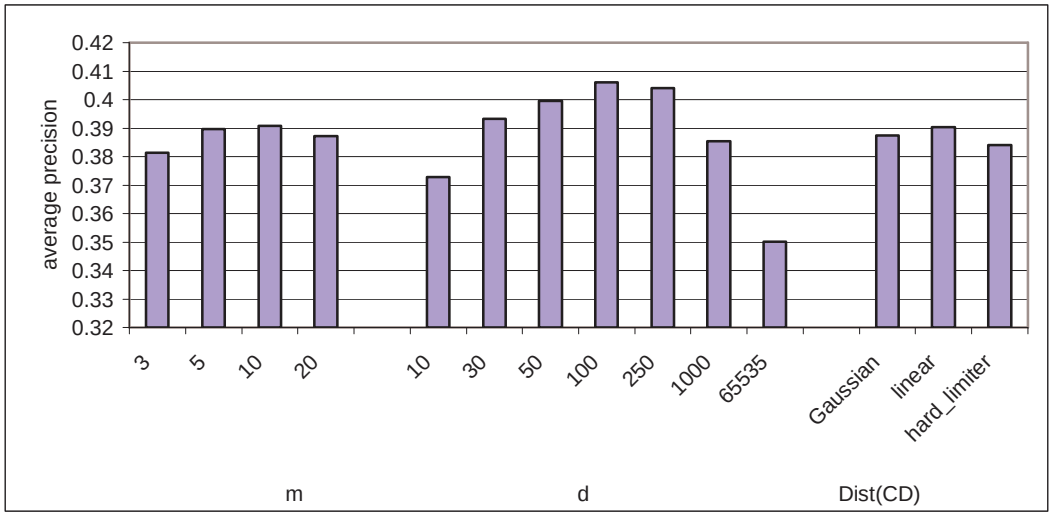

Figure 6. Overall performance for parameters d, m and Dist(CD) across all runs

Taking the combination of parameters m = 10, d = 250 and Dist(*CD*) = linear that yielded the best average precision of 0.4142, we ran some experiments with varying values of n. These results are shown in Table 5. None of the runs though surpassed 0.4142 and this confirms that n = 20 is ideal.

Table 5. Results for different settings of n

n	Avg. Prec.	%Δ	Prec. @ 20	#Rel. Docs
2	0.3850	+28.88%	0.4070	1824
5	0.4070	+36.26%	0.4100	1843
10	0.4107	+37.52%	0.4250	1857
30	0.4004	+34.04%	0.4190	1858
50	0.3916	+31.11%	0.4080	1863

4.5 Determining *w1* and *w2*

w1 and *w2* are set to 0.5 by default. But we wanted to investigate the most ideal settings for both these parameters. Not only are we interested in tuning the system for optimal performance, but *w1* and *w2* give an insight in to the significance of the sub-contexts and of TF vs. CMC respectively. At this stage we had determined that the best parameters were m = 10, d = 250 and Dist(*CD*) = linear and n = 20. Taking these settings, we ran all 121 combinations of 0.1 intervals between [0,1] for *w1* and *w2*. The top 5 resulting runs are shown in Table 6.

Table 6. Top 5 results for combinations of w1 and w2

w1	*w2*	Avg. Prec.	%Δ	Prec. @ 20	#Rel. Docs
0.5	0.4	0.4143	+38.70%	0.4160	1861
0.5	0.3	0.4142	+38.67%	0.4200	1863
0.5	0.5	0.4142	+38.66%	0.4170	1864
0.3	0.5	0.4137	+38.51%	0.4130	1880
0.4	0.5	0.4136	+38.47%	0.4160	1870

These results confirm that when *w1* and *w2* are set to around 0.5, the best results are obtained. A high value for *w1* favours $CI_{q,Q,D}$ over $CI_{q,QR,D}$ (see Equation 3) where as a high value for *w2* favours TF over CMC (see Equation 8).

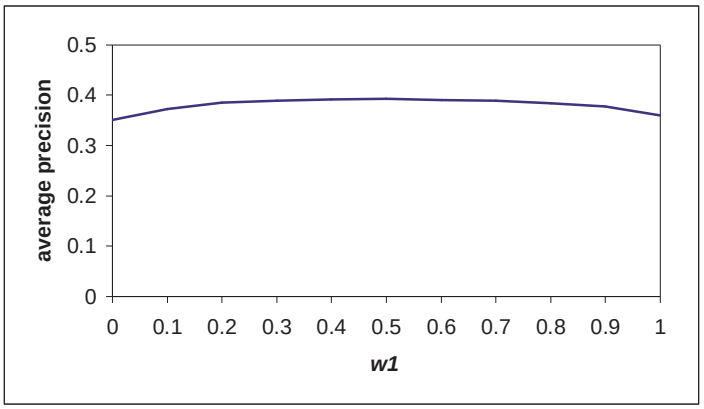

Figure 7. Overall performance of *w1* at intervals of 0.1

Figure 7 shows the average of the average precision across all runs of *w1* at intervals of 0.1. The precision is at a maximum between 0.4-0.6 and at minimums at 0 and 1. When *w1* is 0, this indicates that only CI from *QR* is used in the calculation of CMC. When *w1* is 1, this indicates that only CI from *Q* is used. We can see that the precision of the system is generally better at higher values of *w1*, which indicates that the sub-context *Q* is a slightly more important than sub-context *QR*. This makes sense since *Q* is the original query containing the users information need where *QR* is an artificially estimated extension to the query generated through query expansion.

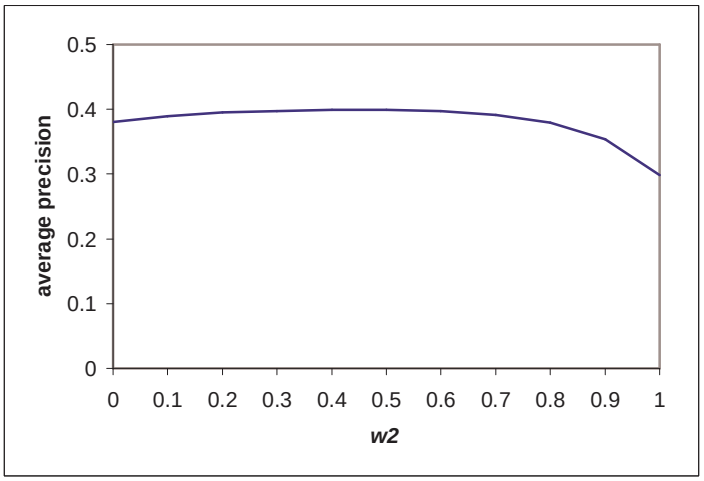

Figure 8. Overall performance of *w2* at intervals of 0.1

Figure 8 shows the average of the average precision across all runs of *w2* at intervals of 0.1. The precision is at a maximum between 0.3-0.6. When *w2* is 1, this only considers TF information in the calculation of TC. When *w2* is 0, this only considers CMC information in the calculation of TC. We can see that CMC alone is an important contributor to the precision of the system with high levels of precision through *w1* = 0 to *w1* = 0.6. We conclude that 0.5 is an acceptable and ideal value for *w1* and *w2* and continue to use this for experiments listed below.

4.6 Determining Relatedness

By default, all terms in the query context *QC* are set a relatedness value of 1. But we experimented with using *IDF* and *TSV* as a measure of relatedness and the results are shown in Table 7.

Table 7. Results for different techniques for obtaining R

R	Avg. Prec.	%Δ	Prec. @ 20	#Rel. Docs
1	0.4142	+38.68%	0.4170	1864
IDF	0.4124	+38.08%	0.4190	1865
TSV	0.4160	+39.27%	0.4200	1863

While *IDF* is more a measure of importance, we interpret it as a type of relatedness measure for this experiment. *TSV* though is a score used to rank potential terms for expansion and makes an ideal type relatedness interpretation. *TSV* gives the best result yielding an average precision of 0.4160, 39.27% better than the baseline run and slightly better than when $R = 1$ or when $R = IDF$. But there is no significant difference between all 3 and we conclude that $R = 1$ as a default value is an acceptable and effective *R* value for all terms.

4.7 Comparison with Previous Results

In this section, we make a final comparison of the performance of the context matching technique against other top performing published results that use the same benchmark data. But making this comparison, we run one final experiment with the ideal parameters $d = 250$, Dist(*CD*) = linear, $m = 10$, $n = 20$, $w1 = 0.5$, $w2 = 0.5$ and $R = 1$. We also substitute *IDF* with the Robertson/Sparck-Jones weight for the weighting of terms in Equation 11 as used in [23]. This experiment yields an improved average precision of 0.4228, precision @ 20 of 0.4380 and 1957 relevant documents. This is an improvement on the previous best observed CM run of 0.4160 and is now a 41.54% gain on the baseline. Table 8 shows the performance of the previous 3 best performing systems with the best CM run in descending order by average precision.

Table 8. CM vs. best previous results

System	Avg. Prec.	Prec. @ 20	#Rel. Docs
Context Matching	0.4228	0.4380	1957
Microsoft (OKAPI/Keenbow) [23]	0.3829	0.4520	2051
Fujitsu Labs [19]	0.3405	0.4010	1988
INQUERY [1]	0.3327	0.4130	1923

Before experimenting with CM on TREC's WT2g benchmark data, the best performing system was OKAPI/Keenbow of Microsoft Research that achieved at average precision of 0.3829 at TREC 8. It is based on OKAPI weighting and also utilizes query expansion. OKAPI weighting was recently successfully used for the content-based retrieval part of topic distillation runs in TREC [8]. The main difference between our context-based result and all the other systems is the context matching technique itself. All the other systems base their document term significance measure only on TF. CM significantly improves retrieval accuracy and outperforms the previous best system of Microsoft by 10.42% with its top performing result of 0.4228. This obviously is a very encouraging and significant result.

The technique seems to be well suited to the nature of the web information retrieval. This is mainly due to the fact the web queries are typically short (2-4 terms) and most documents that are relevant typically contain at least 1 occurrence of the original query term. This means that context matching can effectively boost these types of documents by matching context, rather than relying on the addition of new terms to the original query.

While CM adds no ability for the system to overcome the keyword matching issue, it improves accuracy by being able to determine significance based on context of original query terms occurring in the documents. From this perspective though, the advantage of using CM means that the issue of re-weighting expanded terms and also original query terms can be avoided. In CM, the issue of weighting terms rests in the assignment of *R* values for terms in the context. But as has been shown through experimental results, this is less significant as a default of 1 for all values *R* is just as effective as weighted comparisons.

The combination of CMC with TF (as performed by Equation 8) maintains the perspective of calculating significance through each term in the query as opposed to introducing a global context-based measure of significance for a document. This permits the continued use of known similarity measures such as Equations 9 and 11 and avoids the need of introducing a new combination technique, as would be the case if a global approach was introduced. Thus, CM can be considered as a type of term weighting technique, one that incorporates both TF and context-based information.

4.8 Efficiency

The added complexity of the CM technique results in processing time overheads during retrieval. This is mainly because of the calculation of the closest distance between terms *q* and *c* that must be determined during matching so that the chosen distance function can determine its importance as part of CI calculation.

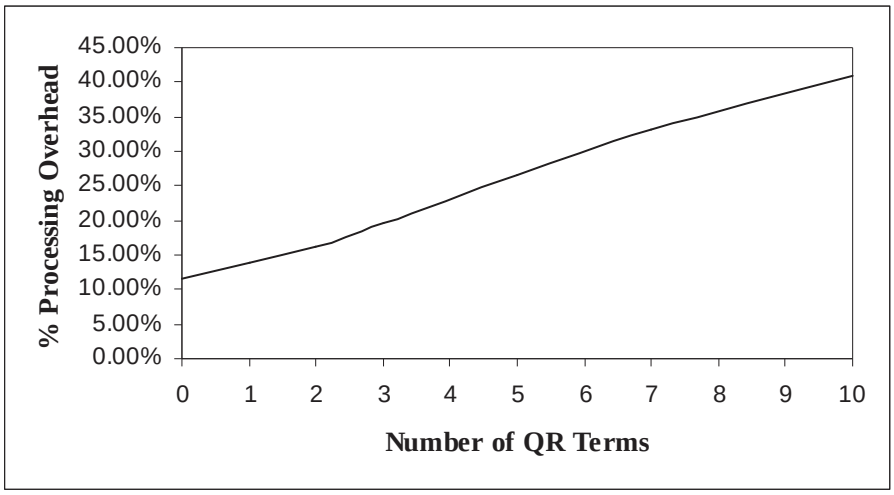

Figure 9. Processing overhead for number of QR terms

Figure 9 shows the online processing overhead at across a different number of *QR* terms. When there are no *QR* terms, then CM is only about 11% slower when used as part of the retrieval process than the baseline run of TFIDF that does not perform CM at all. In this instance, only *Q* is being used as part of the query context *QC* for context matching. This means that only $CI_{q,Q,D}$ is being calculated during CM. That is, only original query terms are being considered as part of the context. But as the number of terms added to *QR* is incremented, the processing overhead increases and follows a linear upward trend. An overhead of approximately 2.5% can be observed for every additional *QR* term that is added. The processing time observations did not measure the time taken for query expansion. The processing time was recorded for just the matching technique itself. This assumes the query context *QC* has already been generated and is available for matching.

The effect on the average precision across the number of *QR* terms is shown in Figure 10. Unavoidably, a trade-off between precision and processing overhead exists that peaks at seven *QR* terms. Additional numbers of *QR* terms beyond seven only result in greater overhead and a degradation in precision.

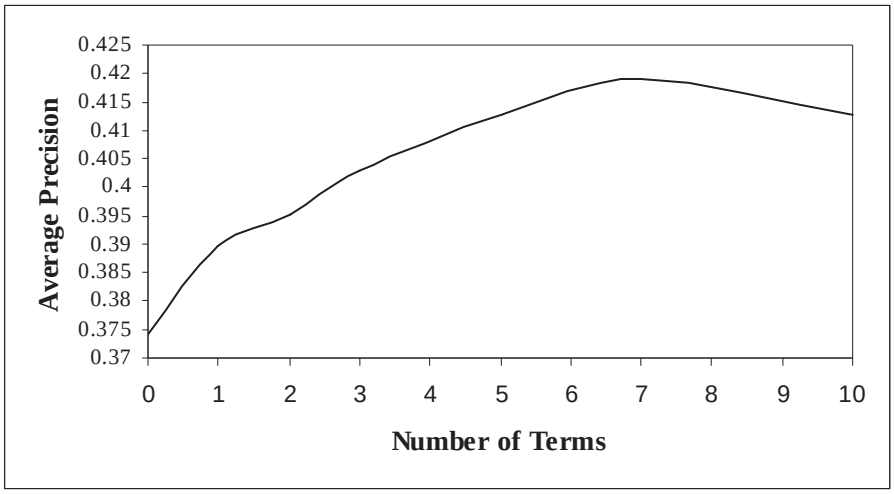

Figure 10. Average precision for numbers of QR terms

Improved efficiency for CM could be achieved by developing advanced functionality in the determination of the closest distance between terms *q* and *c*. At the moment, the positional information stored for each occurrence of a given term in a document is stored as a flat list of integers in memory. So when calculating the closest distance between an occurrence of another term and the list of occurrences of the given term, distance calculations are performed iteratively across every position of the given term in the list to determine the closest distance. This can be thought of as a type of linear searching and is probably the most exhaustive and inefficient method of performing the determination of closest distance for context matching. But on the other hand this type of data structure is easily created and managed during the reading of the term index from file. Other methods such as a binary tree based search may be introduced in the future to improve efficiency.

5 Conclusion

Context continues to proved useful information that can be exploited for effective information retrieval. We have proposed a novel technique called context matching that captures query context and matches this against term contexts in documents to determine term significance and relevancy. CM introduces new ways of interpreting and using context for retrieval and has a significant and positive impact on retrieval accuracy. It has been shown to be a very effective technique by outperforming previous best results by over 10% and the baseline run by over 41%.

We have shown how query context can be formed through original terms and expanded terms obtained from relevance feedback and how this can be effectively used for the context matching. Even when traditional methods of using expanded terms fail to improve retrieval effectiveness, CM can still improve accuracy. While term frequency is a generally a good measure of term significance, it can be combined with CM to boost retrieval effectiveness.

In future research we plan to investigate the use of sentences as a unit of distance and to introduce more advanced matching functions into the process.

References

[1] J. Allan, J. Callan, F. Feng and D. Malin, "INQUERY and TREC-8" in Proceedings of the 8th Text Retrieval Conference (TREC-8), Gaithersburg, USA, pp. 637-643, 1999.

[2] E. Amitay, D. Carmel, A. Darlow, R. Lempel and A. Soffer, “Topic Distillation with Knowledge Agents,” in Proceedings of the 11th Text Retrieval Conference (TREC-11), Gaithersburg, Maryland, USA, 2002.
[3] V. Anh and A. Moffat, "Robust and Web Retrieval Document-Centric Integral Impacts," in Proceedings of the 12th Text Retrieval Conference (TREC-12), Gaithersburg, USA, pp. 726-731, 2003.
[4] R. Baeza-Yates and B. Ribeiro-Neto, "Modern Information Retrieval," Addison Wesley, New York, 1999.
[5] K. Bharat and M. Hezinger, “Improved Algorithms for Topic Distillation in a Hyperlinked Environment,” in the Proceedings of the Annual ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 104-111, Melbourne, Australia, 1998.
[6] H. Billhardt, D. Borrajo and V. Maojo, " A Context Vector Model for Information Retrieval," Journal of the American Society for Information Science and Technology, vol. 53, no. 3, pp. 236-249, 2002.
[7] S. Brin and L. Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine,” In Proceedings of the 7th WWW Conference, pp. 107-117, Brisbane, Australia, 1998a.
[8] N. Craswell, D. Hawking, T. Upstill, A. McLean, R. Wilkinson and M. Wu, "TREC 12 Web and Interactive Tracks at CSIRO," in Proceedings of the 12th Text Retrieval Conference (TREC-12), Gaithersburg, USA, pp. 193-203, 2003.
[9] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman and E. Ruppin, "Placing Search in Context: The Concept Revisited," in Proceedings of the 10th International World Wide Web Conference, pp. 406-414, 2001.
[10] E. Glover, S. Lawrence, M. Gordon, W. Birmingham and C. Lee Giles, "Web Search - Your Way," Communications of the ACM, vol. 44, no. 12, pp. 97-102, 2001.
[11] B. He and I. Ounis, "A Study of Parameter Tuning for Term Frequency Normalization," in Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM), pp. 10-16, 2003.
[12] M. Hezinger, “Link Analysis in Web Information Retrieval,” IEEE Data Engineering Bulletin, vol. 23, no. 3, pp. 38-48, 2000.
[13] T. Honkela, S. Kaski, K. Lagus and T. Kohonen, "WEBSOM - Self-Organizing Maps of Document Collections," in Proceedings of WSOM'97 (Workshop on Self-Organizing Maps), Espoo, Finland, pp. 310-315, 1997.
[14] H. Jing and E. Tzoukermann, "Information Retrieval based on Context Distance and Morphology," in Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in information Retrieval, pp. 90-96, 1999.
[15] I. Kang and G. Kim, "Query Type Classification for Web Document Retrieval," in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval", ACM Press, pp. 64-71, 2003.
[16] J. Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” Journal of the ACM, vol. 46, no. 5, pp. 604-632, 1999.
[17] S. Lawrence and C. Giles, "Context and Page Analysis for Improved Web Search," IEEE Internet Computing, vol. 2, no. 4, pp. 38-46, 1998.
[18] H. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary In formation," IBM Journal of Research and Development, vol. 1, no. 4, pp. 309-317, 1957.
[19] I. Namba and N. Igata, "Fujitsu Laboratories TREC8 Report Adhoc, Small Web, and Large Web Track," in Proceedings of the 8th Text Retrieval Conference (TREC-8), Gaithersburg, USA, pp. 275-284, 1999.
[20] V. Plachouris and I. Ounis, "Query-biased Combination of Evidence on the Web," Workshop on Mathematical/Formal Methods in Information Retrieval, ACM SIGIR Conference, pp. 105-121, 2002.
[21] V. Plachouris, F. Cacheda, Iadh Ounis and C. van Rijsbergen, "University of Glasgow at the Web Track: Dynamic Application of Hyperlink Analysis using the Query Scope," in Proceedings of the 12th Text Retrieval Conference (TREC-12), Gaithersburg, USA, pp. 636-642, 2003.

[22] S. Robertson, "On Term Selection for Query Expansion," Journal of Documentation, vol.46, no. 4, pp. 359-364, 1990.
[23] S. Robertson and S. Walker, "Okapi/Keenbow at TREC-8," in Proceedings of the 8th Text Retrieval Conference (TREC-8), Gaithersburg, USA, pp. 151-161, 1999.
[24] G. Salton and C. Yang, "On the Specification of Term Values in Automatic Indexing," Journal of Documentation, vol. 29, no. 4, pp. 351-372, 1973.
[25] G. Salton, C. Yang and A. Wong, "A Vector Space Model for Automatic Indexing," Communications of the ACM, vol. 18, no. 11, pp. 613-620, 1975.
[26] E. Voorhees, "Using WordNet for Text Retrieval," WordNet: An Electronic Lexical Database, MIT Press, pp. 285-303, 1998.
[27] S. Walker, S. Robertson, M. Boughanem, G. Jones and K. Sparck Jones, "Okapi at TREC-6 Automatic ad hoc, VLC, Routing, Filtering and QSDR," in Proceedings of the 6th Text Retrieval Conference (TREC-6), Gaithersburg, USA, pp. 125-136, 1997.
[28] J. Wen, R.Song, D. Cai, K. Zhu, S. Yu, S. Ye and W. Ma, "Microsoft Research Asia of the Web Track of TREC 2003," in Proceedings of the 12th Text Retrieval Conference (TREC-12), Gaithersburg, USA, pp. 408-417, 2003.
[29] J. Xu and B. Croft, "Query Expansion Using Local and Global Document Analysis," in Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 4-11, 1996.
[30] S. Yu, D. Cai, J. Wen and W. Ma, "Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation," in Proceedings of the 12th International Word Wide Web Conference, 2003.