UTILIZATION OF CROSS-TERMS TO ENHANCE THE LANGUAGE MODEL FOR INFORMATION RETRIEVAL

Huda Mohammed Barakat[1], Maizatul Akmar Ismail[2] and Sri Devi Ravana[3]
[1, 2, 3]Department of Information System, Faculty of Computer Science and Information Technology, University of Malaya,50603, Kuala Lumpur, Malaysia
Email: [1]*eng_huda_barakat@yahoo.com* , [2]*maizatul@um.edu.my* , [3]*sdevi@um.edu.my*

ABSTRACT

Traditional retrieval models were effective in the early stage of the Web; however, with the huge amount of information that is available on the Web today further optimization is required to enhance the performance of these models in extracting the most relevant information. Utilization of the term proximity is one of the techniques that have been introduced for this purpose by many researchers. It assumes that the words in the user query are correlated and thus proximity between them should be considered in the matching process. Density-based proximity is an effectual type of term proximity measures which is still not fully considered in the retrieval models. In this paper we investigate the application of a recent density-based measure called Cross-Terms which has achieved significant scores when applied on the effective BM25 retrieval model. We applied cross-terms on another effective retrieval model that is the Language Modeling Approach. The performance of the enhanced language model was measured and evaluated through several experiments and metrics. Experiments results show that the cross-terms measure was able to improve the performance of the basic language model in all the applied evaluation metrics. Performance improvement reached (+4%) with the MAP metric and (+8%) with P@5 and P@20 metrics.

Keywords: information retrieval, cross-terms, kernel, proximity, language model.

1.0 INTRODUCTION

Today the World Wide Web (WWW) has become a useful resource for variety types of information in different areas. Many retrieval algorithms were developed to handle the process of retrieving and ranking information based on its relevance to the user queries. Traditional retrieval algorithms such as the Boolean retrieval model, the Vector Space model and the Probabilistic Retrieval models [1] were satisfying the users' need of different information in the early stage of the Web. However, with the huge amount of information that we have today on the Web which increases every day, the existing retrieval algorithms need to be more efficient and selective in order to retrieve the most relevant information to the user's searched topic.

Textual information is the main type of information that is available on the Web. There have been lots of efforts to optimize the retrieval algorithms for this type of information including utilization of proximity between query terms in the searched text. In fact, most of the traditional retrieval models treat documents as a bag of words and the retrieval process as a process of matching query terms independently within this bag. However, query terms may have a kind of interdependence among them which when ignored the retrieval results could be irrelevant as shown in the example in Fig.1.In this example we have a query that consists of two correlated words: "operating system", both words appear in two documents (*Doc1, Doc2*) in the data collection with same frequency but with different distances among them in each document. As the example shows traditional retrieval models which score documents based on existence and frequency or existence only of each query term in the document, will assign the same scores to *Doc1, Doc2*. However, *Doc1* is more relevant than *Doc2* as the applied query in fact is about the "operating system" as one notion and not as two separated words.

It can be inferred from the example in Fig. 1 that connection among query terms is reflected by distances between them in each of the documents which contain those terms and that ignorance of these distances when scoring documents could result in irrelevant retrieval results. Clearly, efficient proximity measures among query terms in the scoring process need to be developed and combined with the traditional retrieval models for more accurate retrieval results.

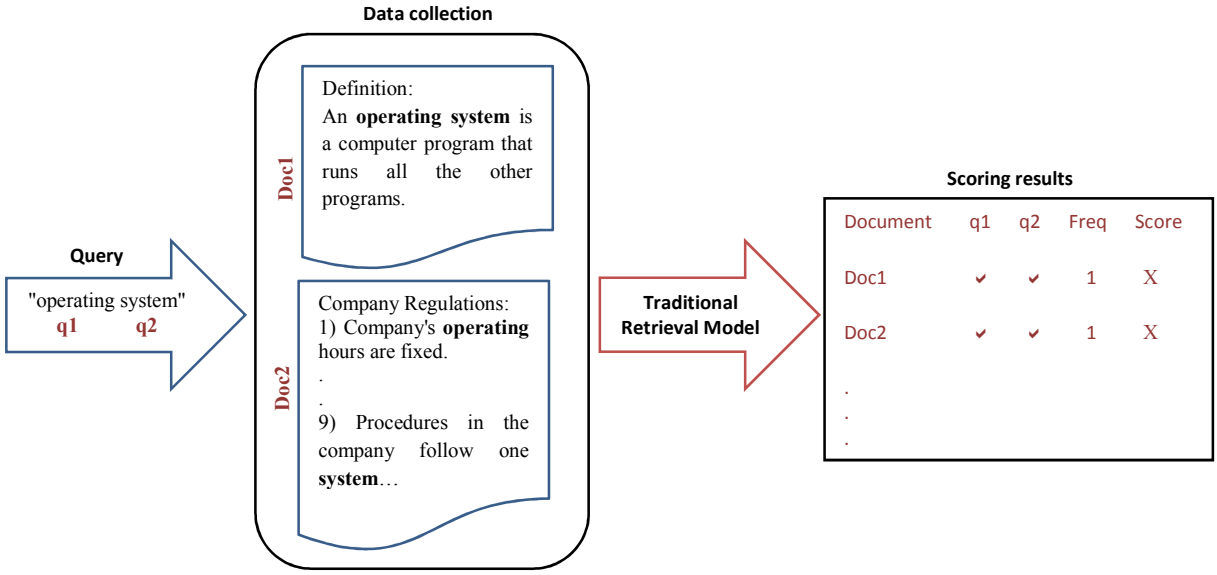

Fig.1:Document scoring by traditional retrieval models

Various term proximity measures have been introduced to improve the retrieval results of several traditional retrieval models. Cross-term proximity measure is one of the recent efforts in this direction. Initial application of this measure was done on the traditional BM25 retrieval model and superior scores over the basic model were accomplished [2]. In this paper, we represent our efforts in applying the cross-terms proximity measure on another efficient and robust retrieval model that is the Language Modeling Approach (LMA) [3]. In fact, LMA has captured the interest of many researchers to investigate and to optimize in the recent years. The significant results that were achieved by LMA in such studies motivated us to use it in our research. To integrate cross-terms into the basic LMA we used a general algorithm known as CRoss TErm Retrieval (CRTER) algorithm [2]. Minor modifications were also introduced to the CRTER algorithm to fit in with the requirements the LMA scoring function.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of the term proximity and prior work that has been done in this direction. In Section 3 we discuss the notion of cross-terms and the CRTER algorithm. Section 4 provides description of LMA which is the base line model for this study. Modified CRTER algorithm that integrates cross-terms into the language model is described in Section 5. We than explain the main processes in experimental setup in Section 6 followed by experiments results and performance evaluation of the proposed model in Section 7. Finally, we conclude our work and give some suggestions for the future work in Section 8.

2.0 RELATED WORKS

According to Hawking and Thistlewaite [4] distance through which proximity among terms can be captured, has three meanings including:

- Term-vector distance
  A good example of the term-vector distance is the SMART system [5] in which documents and queries are represented as vectors of the terms' weights. The weight of the term in this identification of the distance is an indication to the frequency of the term occurrence within the document.

- Conceptual distance

The conceptual distance is based mainly on the *concepts maps* and has been applied in several proximity approaches such as the popular INQUERY system [6], Markov Random Field Model [7] and many other approaches.

- Lexical distance
  In the Lexical distance, terms proximity is measured using the number of words located between any two query terms in the document, [8]. In fact, cross-terms measure which is considered in this paper can be classified under this type of term proximity measures. Based on the reviewed works in this study it was found that most of the lexical measures use one of three main methods to calculate distance between any two query terms in the document including span-based methods, pair-based methods or density-based methods. It can be also concluded that these methods are either independent retrieval models or dependent (more general) functions that are applied on top of the retrieval results of a traditional retrieval model or integrated into it.

2.1 Span-Based Proximity

Proximity based on spans scores documents by segmenting each document into spans (parts of text) before weighing each span based on the query terms that it covers. The whole document is then scored and ranked according to the summation value of its spans' weights. Shortest substrings model [9] and Z-mode [4] are two early span-based approaches which were introduced in the same year. Both approaches are built on similar assumptions and mainly on scoring documents by looking for the shortest spans. The shortest substrings model was further extended in the Cover Density Ranking approach [10] which applied Boolean queries to include unstructured short queries that do not exceed three words. Song and his group mentioned a main defect in the previous two models (Shortest substrings model and Z-mode model) which is the exclusion of the traditional ranking functions, i.e., *tf.idf (term frequency. inverse document frequency)* and the document length [11]. They proposed an approach that overcame this defect and also another two problems in span-based approaches including creation of non-overlapping spans and consideration of term context when measuring its contribution to the document relevance[11].
Other span-based approaches search for the minimum interval that contains all *k* keywords in each document as in Plane-sweep and Divide-and-Conquer algorithms. Specific sorting for keyword positions can be also considered as in the former algorithm [12]. Recently, several span-based measures were proposed and evaluated in two useful works [12, 13]. The work of Tao and Zhai [14] is more comprehensive than Cummins & O'Riordan's work [13] as they discussed 12 different proximity measures and evaluated integration of some of these measures into one proximity function. The resulting measures were effective for both long and short queries while Cummins and O'Riordan's work [13] was limited to short queries only.

2.2 Pair-Based Proximity

Proximity based on term pairs (also called distance aggregation) measures the pair-wise distance between query terms in the document and then score it by aggregating these distances. Primitively early efforts under this category of proximity measures appeared in various studies such as [15], [16] where seeking relevant documents is based on matching queries with only two keywords. Manber and Baeza–Yates's work [17] is more advanced as they were able to retrieve a tuple of keywords using a repeated process where a keyword is added in each cycle until all the tuples have been covered.
Other direction of efforts under this category is to apply a proposed pair-based measure on the top (X) retrieved results by a traditional retrieval model. For instance, in Rasolofo and Savoy's study [18],Okapi's top 100 documents were ranked by an additional pair-based proximity measure. One main shortage in this approach is the linear combination of phrases and single terms in the scoring function[11].

2.3 Density-Based Proximity

Proximity based on density or distribution uses mathematical functions (kernel functions) to represent the density of each term (Kernel) within the whole document. This type of proximity measures assumes that the occurrence of each query term in the document has an influence on its neighboring text which decreases as we go far from that

term. Different relationships can be then defined between the terms' influences and consequently utilized in building proximity measures among these terms within the document.
Kretser & Moffat's work [19] is one of the earliest studies that used a density-based approach to obtain terms proximity. The spread of the query terms influences within the collection were assumed to be additive at any position in the collection causing either high or low peaks at that position. High density of query terms is detected by high peaks and thus the text centered by such peaks is considered more relevant to the query and vice versa.
Petkova and Croft [20] also used a kernel-based representation for documents to enhance another type of text retrieval named *Entity Retrieval*. Proximity here is not between query terms but between entities, which are pre identified concepts in the document, and words (references) surrounding each entity.
Recently, the Positional Language Model (PLM) [21] and the CRossTErm Retrieval (CRTER) model [2] are two new models that apply density-based proximity measures. As its name indicates, PLM captures terms proximity through the language models that are identified for each position in the document using propagation (kernel) functions. In CRTER model term proximity is determined in different way using new pseudo terms named cross-terms. CRTER was initially examined on the BM25 model and was able to achieve high retrieval results. In this paper, will be applying CRTER on another effective retrieval model, i.e., the language modeling approach.

3.0 CRTER MODEL

The CRossTErm Retrieval model (CRTER) is a general retrieval model that introduces a density-based proximity measure called Cross-Terms [2].

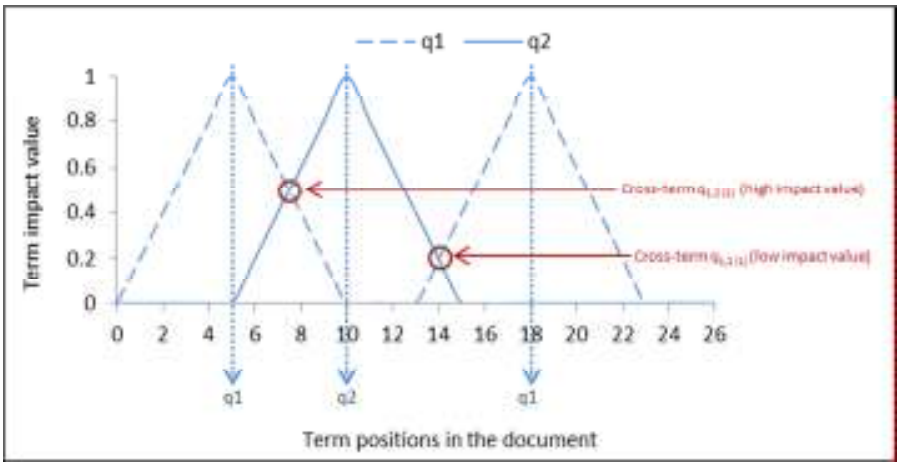

Fig.2: Example of Cross terms

The proposed proximity measure in this model can be integrated into any traditional retrieval model through CRTER's general algorithm. As being a density-based measure, cross-terms assumes that the occurrence of each query term in a specific position in the document has an impact on its neighboring text which decreases as we go far from this position. Therefore this impact is represented by a distribution kernel function which equals to 1 at the term occurrence point and become less than 1 as it moves to positions that are far from this point. A cross-term then occurs when the impact shape functions of two query terms (kernels) within the same document intersect in one or more points. The intersection value is called cross-term and is inversely proportional to the distance between the query terms forming it. Fig.2 shows different examples of cross terms for two query terms (q1, q2).

As Fig. 2 illustrates, q_1 appears at two positions in the document (5, 18) while q_2 appears once in the document at position (10). The impact shape functions of the two terms (q_1, q_2) intersect at two positions forming two cross

terms $q_{1,2(1)}$ and $q_{1,2(2)}$. Obviously, the value of cross term $q_{1,2(1)}$ is higher than $q_{1,2(2)}$ because the query terms forming$q_{1,2(1)}$ are closer to each other in the document than those forming $q_{1,2(1)}$.

The scoring function of CRTER is a linear combination of two parts each of them is a copy of the scoring function of the model on which CRTER is applied. In the first part, the variants of the scoring function are calculated using ordinary query terms while the second part uses query cross-terms instead. Obviously, term proximity is involved into the score via second part of the function. Equation (1) shows CRTER's general scoring function.

$$\mathrm{CRTER(d)} = (1-\lambda).\sum_{1\le i\le k} w(q_i,d) + \lambda.\sum_{1\le i<j\le k} w'(q_{ij},d) \qquad (1)$$

Where w is the applied weighting function with query terms q_i and w' is the same weighting function (w) but with cross-terms $q_{i,j}$. Parameter λ is a balancing factor between query terms and query cross-terms so that when λ=0, the retrieval model uses query terms only and becomes the standard weighting function of the applied model. However, when λ=1, CRTER will be using the weighting function of the applied model with cross-terms of the query only.

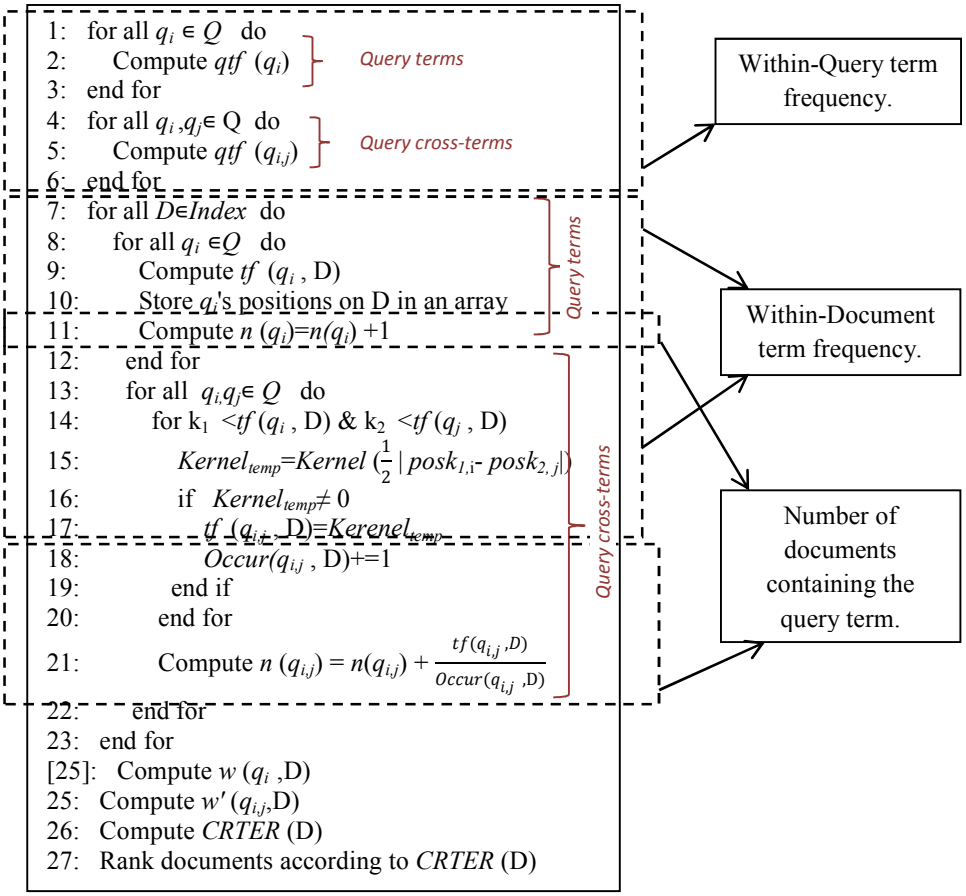

Fig.3: Main parts of CRTER original algorithm [2]

Cross-terms were first applied on the BM25 retrieval model and thus variants of its scoring function were considered in the original CRTER algorithm as shown by Fig. 3. The scoring function of BM25 has three main variants including the within-document term frequency, within-query term frequency and number of documents that contains the term. In CRTER algorithm, the three variants were calculated directly with ordinary terms of the

query. However, with the pseudo terms (cross-terms) there were specific equations developed by CRTER founders to calculate each of the previous variants and these are shown in Equation (2),(3) and (4).

$$\text{With-in Document Frequency} = tf(q_{i,j}, \mathrm{d}) = \sum_{k_1=1}^{tf_i} \sum_{k_2=1}^{tf_j} Kernel\left(\frac{1}{2}\, dist\left(pos_{k1,i}, pos_{k2,j}\right)\right) \quad (2)$$

$$\text{With-in Query Frequency} = qtf(q_{i,j}, d) = Kernel\left(\frac{1}{2}\right).min\left(qtf(q_i), qtf(q_j)\right) \quad (3)$$

$$\text{Number of documents containing the term } \mathrm{q_{i,j}} = n(q_{i,j}) = \sum_{D \in Index, Occur(qi,j)\neq 0} \frac{tf(q_{i,j})}{Occur(q_{i,j}, D)} \quad (4)$$

Where *Kernel* is the applied distribution function which can be one of seven functions including Gaussian, Triangle, Circle, Cosine, Quartic, Epanechnikov and Triweight. *Tf* and *qtf* are the occurrence values of cross-term $q_{i,j}$ within the document and within the query respectively. *Index* is the data collection and *occur*($q_{i,j}$, *D)* is calculated as shown in Equation (5).

$$Occur(q_{i,j}, \mathrm{d}) = \sum_{k_1=1}^{tf_i} \sum_{k_2=1}^{tf_j} 1_{\left\{x, Kernel\left(\frac{1}{2} dist\,(posk1,i, posk2,j)\right)\neq 0\right\}} \quad (5)$$

Where tf_i and tf_j are frequencies of terms q_i and q_j in document *d*, *posk1,i* and *posk2,j* are the positions of occurrences *k1* and *k2* of terms q_i and q_j in document *d* respectively. Function *dist* calculates the distance between the two positions *posk1,i* and *posk2,j* and *kernel* is the applied kernel function.

4.0 THE LANGUAGE MODELING APPROACH

The term "Language Model" for any text or sequence of words refers to the distribution of probability among these words (or any text units). In the Information Retrieval (IR) area, language model is utilized in the retrieval process through two steps. First, a language model is estimated for each document in the collection. Second, the estimated model is used to calculate the probability of generating the query by each document where the calculated values are used to score and rank documents [22]. Generally, these two steps are known as the Language Modeling Approach (LMA) which is still considered a new approach in informational retrieval compared to the other traditional retrieval models. LMA was first introduced in this area by Ponte and Croft in 1998 and known as the *query likelihood model* which is the basic form of LMA [3].

At nearly the same time of introducing the query likelihood model there was another similar model developed by two other research groups. The model is known as the *unigram language model* or the *multinomial model* [23], [24]. In fact, the final scoring functions of both models have the same general form which consists of two main parts. The first part is the *local* part where probability is related to the document only. The second part is known as the *global* part and probability in this part is calculated based on the entire collection. However, the two models differ in how both of these components are estimated in each model. The global part is also known as collection part and its main role in the scoring function is to avoid Zero document probability that can be caused by query terms that do not appear in that document [25],[21].

In this study, the CRTER model was implemented with one form of basic language model that is the unigram language model [24]. In fact, this form of the language model has captured the interest of many researchers since it was developed and until today. Scoring document in the unigram language model is done by a multinomial distribution of probability which is simply a multiplication of the maximum likelihood of every query term within the document. Therefore, the probability of generating a query Q= [t_1,t_2,t_3,…,t_n] from a document *d* is calculated by the following equation:

$$P(Q \mid d) = \prod_{i=1}^{n} P(t_i|d) \qquad (6)$$

Where $P(t_i|d)$ is the maximum likelihood of term t_i in document d. For the collection part, the model uses the term t_i frequency over the entire collection $tf(t_i)$ divided by the total number of terms in the collection $tf(t)$. The final form of the scoring function in this model uses a linear combination of the previous two parts as shown in Equation (7).

$$P(Q \mid d) = \prod_{i=1}^{n} \left((1-\propto) \frac{tf(t_i,d)}{|d|} + \propto \frac{tf(t_i)}{tf(t)} \right) \qquad (7)$$

Equation (7) is the equation that has been used in this study to represent the base line model.

5.0 THE UNIGRAM LANGUAGE MODEL WITH CRTER

As we mentioned before, the proposed model in this study is to integrate cross-terms into the unigram language model [24] using the CRTER algorithm. The scoring function of the unigram model contains three main variants including: (i) query term frequency within the document, (ii) query term frequency within the collection and (iii) entire number of terms in the collection. Referring to the original CRTER algorithm (see to Fig.3) there are calculations for some variants that are not used by the scoring function of the unigram model. Specifically, lines for computing the within-query term frequency and the number of documents that contain the query term were not applicable to the unigram language model and thus had to be removed from the algorithm in this case.
In addition, the third variant in the scoring function of the unigram model does not exist in the original CRTER algorithm and thus had to be added, i.e., query term frequency within the entire collection (refer to Equation (7)).
As mentioned earlier, each variant in the applied scoring function in CRTER's main equation is calculated twice; one by using ordinary terms of the query and the second is by using cross-terms of the query. In our case, calculation of the above three variants using the ordinary terms of the query can be done by direct counting of: each query term within each document, each query term within the collection and all terms in the collection respectively.
However, implementation of the first two variants using query cross-terms needs specific equations that take into consideration the value of each cross-term occurrence. Equation (2) which is developed by founders of CRTER is used to calculate the cross-term frequency within the document. Another equation is added to the CRTER algorithm in this study to compute the query cross-term frequency within the collection and it is shown by Equation (8). In fact, Equation (8) depends on Equation (2) and both are implemented in steps (7-15) in the modified CRTER algorithm.

$$tf_{Col-cross}(q_{i,j}, d) = \sum_{k_1=1}^{tf_i} \sum_{k_2=1}^{tf_j} tf_{doc-cross}(q_{i,j}, d) \qquad (8)$$

Where $tf_{doc\text{-}cross}(q_{i,j}, d)$ is calculated by Equation (2). The final CRTER algorithm after modification (deletion and addition) will be as shown in Fig.4. To calculate the third variant (number of terms in the collection) with cross-terms we just used a direct counting of terms in the collection as cross-terms is not applicable to this variant.
As for the time complexity of the modified CRTER algorithm it was found to be the same as the original algorithm [2]. Equation (9) shows time complexity of the proposed model.

$$O(\ |\text{index}|.(\ |Q|.|D|\ )\ ) \qquad (9)$$

Where index is the collection size, |Q| is the query length and |D| is the average length of document in the index.

6.0 EXPERIMENTAL SETUP

6.1 Data Collection

All the testing experiments of the proposed model in this study were done on a small data collection which is the CACM free data collection. The CACM collection comes from the Association of Computing Machinery (ACM) and it is meant for research purposes only. The collection consists of abstracts of articles published in the period

between 1958 and 1979 in the Communications of the ACM journal. Many IR related studies have used the CACM collection which consists of (30[25]) documents and (52) queries. Relevance judgments are also provided with the collection for the evaluation process [26].

6.2 Base Line Specification

This research aims at integrating cross-terms proximity measure into a basic form of the LMA where the unigram language model [24] has been chosen as the base line model. Before starting the testing process of the optimized model, performance of the base line model had to be tested in order to identify the best value for its smoothing parameter ($\propto$) (refer to equation (7)). As parameter $\propto$ takes values between (0,1) the base line model was examined on 9 different values for $\propto$ where the best Mean Average Precision (MAP) score was obtained with $\propto$=0.9. MAP is the main evaluation metric used in this study.

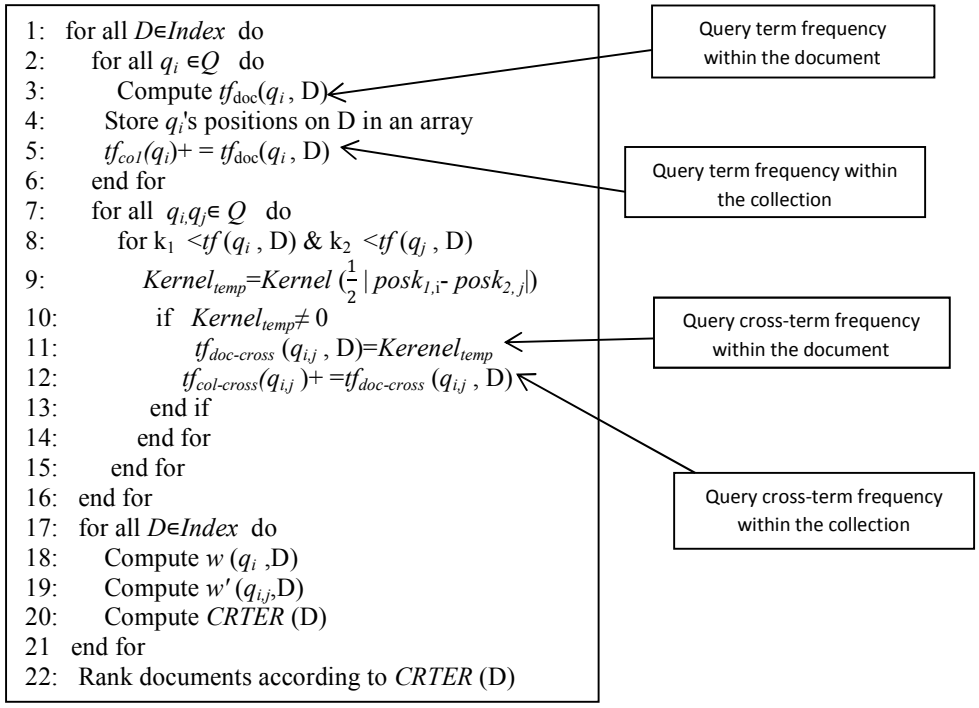

Fig.4: CRTER algorithm for the unigram language model.

6.3 Kernel Functions

The kernel functions are used to represent the impact shape of each query terms occurrence in the documents and capture the intersection value (cross-term) between different query terms within the document [2]. The seven kernel functions that were used by founders of CRTER were also implemented in this study. The following are the seven equations of these seven applied kernels.

$$\text{Kernel} - \text{Gaussian}(x) = \exp\left[\frac{-x^2}{2\sigma^2}\right] \quad (11)$$

$$\text{Kernel} - \text{Triangle}(x) = \left(1 - \frac{x}{\sigma}\right) . 1_{\{x \leq \sigma\}} \quad (12)$$

$$\text{Kernel} - \text{Circle(x)} = \sqrt{\left(1 - \frac{x}{\sigma}\right)^2} \quad .1_{\{x \le \sigma\}} \tag{13}$$

$$\text{Kernel} - \text{Cosine(x)} = \frac{1}{2}\left[1 + cos\left(\frac{x\pi}{\sigma}\right)\right].1_{\{x \le \sigma\}} \tag{14}$$

$$\text{Kernel} - \text{Quartic(x)} = \left(\left(\frac{x}{\sigma}\right)^2\right)^2 .1_{\{x \le \sigma\}} \tag{15}$$

$$\text{Kernel} - \text{Epanechnikov(x)} = \left(1 - \left(\frac{x}{\sigma}\right)^2\right).1_{\{x \le \sigma\}} \tag{16}$$

$$\text{Kernel} - \text{Triweight(x)} = \left(1 - \left(\frac{x}{\sigma}\right)^2\right)^3 .1_{\{x \le \sigma\}} \tag{17}$$

Where σ is a normalization parameter that determines the impact range of the term (curve expansion), $.1_{\{x \le \sigma\}}$ is an indicator function which equals 1 when $x \le \sigma$otherwise it equals 0 and thus the whole function will equal 0.

6.4 Parameters File

The proposed model in this study contains four main parameters: the first three parameters are Lambda (ʎ), kernel function (*kf*) and Sigma (σ) which are used in the CRTER main equation. The ʎ parameter can be seen directly in CRTER's main equation, while *kf* and σ parameters (named kernel parameters) exist implicitly in the weighting function that is located in the second part of the same equation (i.e., *w'*refer to equation (1)). The fourth parameter is Alpha (∝) and this is the smoothing parameter in the applied equation of LMA and which is shown by Equation (7).
To evaluate the performance of the proposed model in this study several experiments were conducted with different values for every parameter in each experiment. Therefore, a parameters file (*CRTER-Eval.param*) that contains these four parameters was created and through this file parameters' different values were passed to the model in each single experiment. Fig shows part of the parameters file that was used in the experiments.

```
<!--  Language model smoothing parameter  -->
<LMLambda>0.9</LMLambda>

<!--  Parameters for the CRTER  -->

<CRTERSigma>1</CRTERSigma>

<!-- Kernel function: 1: Gaussian; 2: Triangle; 3: Circle 4: Cosine; 5: Quartic;
6: Epanechnikov; 7: Triweight -->
<CRTERKF>2</CRTERKF>

<CRTERAlpha>0.1</CRTERAlpha>
```

Fig. 5: Part of the parameters file used in testing the proposed model

6.5 Supporting Applications

The proposed model was developed using the C++ programming language (MS Visual C++ 2008) with the support of an open source toolkit named *Lemur* version (4.12). Lemur toolkit is a collection of software tools and search engines which is mainly designed to support researches using statistical language models for different information retrieval tasks. The toolkit was developed as a first product by the Lemur project group who modifies and enhances it periodically [27].

As some files in Lemur toolkit that were integrated into the developed model in this study were based on Linux platform, another open source toolkit named Cygwin were also used in the model implantation process. Cygwin is a collection of tools that provide the Linux environment on Windows platform [28].

6.6 Evaluation Metrics

For model evaluation two main TREC official evaluation metrics [29] were applied as follows:

- **Mean Average Precision** (MAP): Precision in IR means the fraction of the retrieved documents that are relevant. Let A be the set of documents in the data collection. Now if R is the set of retrieved documents by query q and R' is the set of relevant documents in R, than precision P(q) is calculated as follows:

$$P(q) = \frac{R\prime}{R} \qquad (18)$$

  Average precision computes precision at every position in the ranked sequence of documents then calculates the average of the resulting precisions. Mean average precision for a set of queries, is the mean of the average precision scores for each query.

- **(P@5) and (P@20):** these two evaluation measures are used to emphasize more on the top retrieved documents. In these measures the precision is re-calculated at the top (x) retrieved documents where x in (P@5) equals 5 and in (P@20) it equals 20 of the top retrieved documents.

7.0 RESULTS & DISCUSSION

CRTER had three main parameters and thus to be able to obtain the best performance of the proposed model in this study, it had to be examined for each fixed value of every parameter over all the possible values of other parameters. For instance, for each value of parameter λ the model needed to be examined for all possible values of parameter *kf* which corresponds to the applied kernel functions in the developed model. Similarly, for each value of the second parameter *kf* the model performance had to be recoded over all the possible values for the third parameter, i.e., σ**.**

While second parameter *kf* has seven fixed values (the applied kernel functions), the values' range of the other two parameters was determined based on the model performance. Experiments showed that the model performance drops down as λ gets larger and starting from λ=0.4 most of the results were below the base line. Similarly, the model score was decreasing as the value of parameter σ becomes larger until parameter σ =40 and onwards where score starts to drop down below the base line.

Accordingly, the model results were recorded with seven values for parameter *kf*, four values for parameter λ namely {0.1,0.2,0.3,0.4} and 15 different values between 1 and 40 for parameter σ. It should be noticed that the increment in the applied values of parameter σ were initially equal to 1 until σ =10 then the increment became larger as the model was more sensitive to σ's small values. Therefore the set of the applied values for parameter σ included {1,2,3,4,5,6,7,8,9,10,15,20,25,30,40}.

7.1 Performance Improvements

Experiments showed that the proposed model (unigram language model with cross-terms) generates high scores for the three applied evaluation metrics when parameter λ is located between 0 and 0.2. It was also observed that values in the range {2, 10} for parameter σ give high scores for all the evaluation metrics. In addition, no one single kernel function in model testing was able to outperform all other functions over the three evaluation measures. The Gaussian kernel, however and according to Table 1 obtained the best scores for both metrics MAP and P@20. For P@5 metric there was no unique superior score as four kernel functions including Circle, Cosine, Quartic and Epanechnikov have achieved the same highest score. Improvement percentages over the base line scores for each of the best scores of the proposed model are also showed in column (+BL) in Table 1. Besides, best scores that our

model achieved for each metric are shown in bold font in the same table.
Regarding scores in Table 1 it is obvious that (*kf*=Gaussian, ⋋=0.1andσ=2) are the best values for the three parameters that have achieved the highest MAP and P@20 scores. As for P@5 metric, these three parameters values were also able to generate a significant score which is slightly smaller than the highest score that this metric has achieved. Accordingly, (*kf*=Gaussian, ⋋=0.1 and σ=2) have been considered as the optimal values for the proposed model parameters based on the applied data collection.

Table 1: Best scores for each kernel with the related values of parameters ⋋ and σ

Kernel	Σ	⋋	MAP	+BL	Σ	⋋	P@5	+BL	Σ	⋋	P@20	+BL
Gaussian	2	0.1	**0.3183**	+4.3984%	1	0.1	0.4385	+7.9133%	2	0.1	**0.2529**	+8.3828%
Triangle	3	0.1	0.3162	+3.7634%	2 3	0.1 0.2	0.4385	+7.9133%	2	0.2	0.251	+7.6892%
Circle	3	0.1	0.3176	+4.1877%	3	0.1	0.4423	**+8.7045%**	3 5 6 2 3	0.1 0.2 0.3	0.2500	+7.3200%
Cosine	3	0.2	0.3169	+3.9760%	3	0.2	0.4423	**+8.7045%**	2	0.2	0.2519	+8.0191%
Quartic	3	0.1	0.3169	+3.9760%	3	0.1	0.4423	**+8.7045%**	3 2 5 6	0.1 0.2	0.2500	+7.3200%
Epanechnikov	3	0.2	0.3167	+3.9154%	3	0.2	0.4423	**+8.7045%**	5	0.2	0.2500	+7.3200%
Triweight	3	0.1	0.3174	+4.1730%	2 3 4	0.1	0.4385	+7.9133%	2 5	0.2	0.2510	+7.6892%

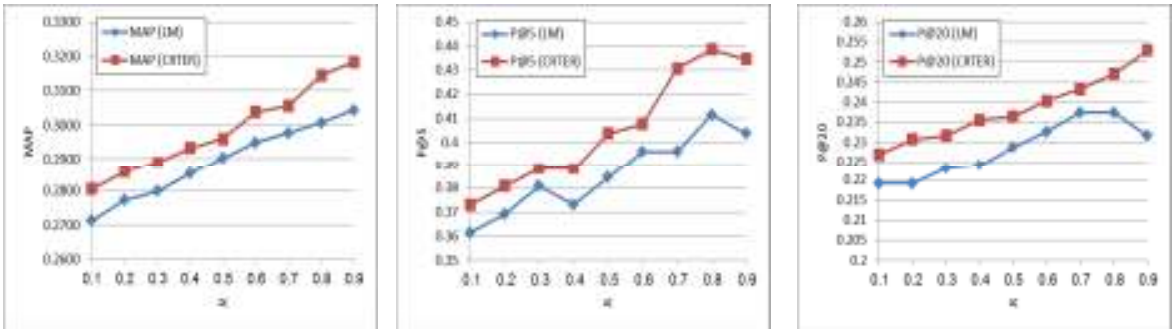

Fig.5:Performance comparison of CRTER (unigram language model with cross-terms) and the basic unigram language model over different values of parameter ∝ (using the optimal values of CRTER parameters)

However, after identifying the empirical optimal values for the proposed model we re-examined the model but with different values for parameter $\propto$ while all parameters of the model were set to their optimal values. Results of re-examination have ensured the superiority of the enhanced unigram language with cross-terms over the original unigram model.

Fig.**5** shows these results in charts which illustrate clearly how CRTER with the unigram language model outperforms the basic unigram model over all values of parameter $\propto$ in all evaluation metrics.

7.2 Parameters Preferred Values

As experiments results illustrate, the proposed model is sensitive to the values of its different parameters. Starting with parameter λ, CRTER scoring function uses a linear incorporation of cross-terms as a proximity measure into the scoring function of the applied model where parameter λ balances the two parts of the equation (see Equation (1)). In fact, the values of λ that are located between 0 and 1 reflect the influence size of cross-terms incorporation into the basic language model. Similar to results achieved by CRTER with BM25 model [2], performance of proposed model in this study decreases as λ gets larger. The fallback in the model performance can be interpreted as a result of reducing the role of the original model scoring function on the account of increasing the influence of cross-terms in CRTER general scoring function. Accordingly, this study concludes that cross-terms influence should be limited to an assisting element in the model scoring function without exceeding it.

Moving to CRTER second parameter, i.e., the kernel function parameter (*kf*), it was seen throughout the results that no one specific distribution function was able to outperform all other functions in all the three evaluation metrics. The BM25 model with CRTER [2] also arrived at the same conclusion; however, the proposed model in this study still needs to be examined on larger data collections to generalize this conclusion.

The kernel function performance is strongly related to its parameter σ and consequently the final retrieval results of the model which is related to this function. Parameter σ controls the extension of the kernel curve and thus its optimal values depend on the query terms distribution within the document. The proposed model was tested through a wide range of σ's values from 1 to 40. At first the performance of the model was increasing as σ's value increases, this is obviously due to the effect of term proximity integration. However, as σ gets larger the model performance started to drop down. σ's large values means wider range of the influence shape of each query term occurrence in the document and thus more cross-terms which consequently adverse their affect in improving the performance. In fact, the influence of parameters *kf* and σ is connected to the applied data collection as their influence is based on the distribution of the query terms in the documents of the collection.

The final parameter which has an indirect influence on the proposed model performance is parameter $\propto$. The role of this parameter is to control the effect of the collection part in the scoring function of the basic language model (refer to equation (7)). In spite of the optimal value of this parameter that we used in our experiments, additional experiments with optimal values of the proposed model parameters have proven superiority of its performance over different values of parameter $\propto$. Therefore it can be concluded that parameter $\propto$ does not have a direct influence on the proposed model performance but its affect is on the retrieval scores of the basic language model which in turn affects the overall performance of the proposed model.

8.0 CONCLUSION AND FUTURE WORK

In this paper we have introduced our efforts to enhance the language modeling approach for information retrieval via utilization of a new density-based proximity measure namely Cross-Terms. Founders of this effective measure have developed a general algorithm known as the CRossTErm Retrieval model or shortly CRTER model that can be used to integrate cross-terms into any traditional retrieval model. We first described some modifications that we made on the original algorithm of CRTER to fit in with the scoring function of the applied language model. Then we discussed the process of testing and evaluating the proposed model performance. Experimental results showed a significant performance of the proposed model with different evaluation metrics which reached (+4%) with the MAP metric and (+8%) with P@5 and P@20 metrics. These results show clearly the effectiveness of the density-

based proximity measures in general and cross-terms proximity measure specifically in enhancing retrieval results of the traditional retrieval models. Further evaluation of the proposed model on larger and standard data collections is one pressing future work. Another direction for the future work is the incorporation of cross-terms into other forms of the language modeling approach (such as KL-divergence retrieval model) which have better performance over the model basic forms and thus better retrieval results is also expected from such incorporation.

ACKNOWLEDGEMENT

Thanks to the all the people who supported this research from the Information Systems Department at University of Malaya. Special thanks go to Dr. Maizatul Akmar Ismail the research supervisor and Dr. Sri Devi Ravana for her guidance in writing this paper.

REFERENCES

[1] C. Manning,P. Raghavan, & H. Schütze,*An Introduction to Information Retrieval*, Cambridge, England, Cambridge University Press, 2009.

[2] J. Zhao,J. X. Huang, B. He,"CRTER: using cross terms to enhance probabilistic information retrieval*",* in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, 2011.

[3] J. M. Ponte, W. B. Croft, "A language modeling approach to information retrieval", in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 1998.

[4] D. Hawking, P. Thistlewaite, "Relevance weighting using distance between term occurrences",*Computer Science Technical Report TR-CS-96-08, Australian National University*, 1996.

[5] C. Buckley, G. Salton, J. Allan, "Automatic retrieval with locality information using SMART*",* in *the First Text Retrieval Conference (TREC-1)*, U.S. National Institute of Satndered and Technology, 1992.

[6] J. P. Callan, W. B. Croft, S. M. Harding, "The INQUERY retrieval system*",* in *the 3rd International Conference on Database and Expert Systems Applications*, 1992.

[7] D. Metzler, W. B. Croft, "A Markov random field model for term dependencies*",* in *the Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005.

[8] R.G. Raj, V. Balakrishnan. "A Model for Determining The Degree of Contradictions in Information". Malaysian Journal of Computer Science, 2011, 24(3): 160 – 167.

[9] C. L. A. Clarke, G. Cormack, F. Burkowski, "Shortest substring ranking (MultiText experiments for TREC-4).,*Paper presented at the 4th TREC*, 1996.

[10] C. L. A. Clarke, G. V. Cormack, E. A. Tudhope,"Relevance ranking for one to three term queries",*Information Processing & Management, 36*(2), pp. 291-311, 2000.

[11] R. Song, J. R. Wen, W. Y. Ma, "Viewing Term Proximity from a Different Perspective", *Advances in Information Retrieval. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven & R. White (Eds.)*, Vol. 4956, pp. 346-357: Springer Berlin / Heidelberg, 2008.

[12] K. Sadakane, H. Imai,"Text Retrieval by using k-word Proximity Search*",* in *Proceedings of International Symposium on Database Applications in Non-Traditional Environments (DANTE'99)*, 1999.

[13] R. Cummins, C. O'Riordan, "Learning in a pairwise term-term proximity framework for information retrieval", in *the Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009.

[14] T. Tao, C. Zhai,"An exploration of proximity measures in information retrieval*"*, in *the Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007.

[15] W. G. Aref, D. Barbara, S. Johnson, S. Mehrotra, "Efficient processing of proximity queries for large databases".in *the Proceedings of the Eleventh International Conference on Data Engineering*, 1995.

[16] G. H. Gonnet, R. A. Baeza-Yates, "NEW INDICES FOR TEXT: PAT TREES AND PAT ARRAYS", in R. A. B.-Y. Gaston H. Gonnet , Tim Snider (Ed.), *Information Retrieval: Data Structures & Algorithms* (pp. 66-82): Prentice Hall, 1992.

[17] U. Manber, R. Baeza–Yates,"An algorithm for string matching with a sequence of don't cares", *Information Processing Letters, 37*(3), 133-136, 1991.

[18] Y. Rasolofo, J. Savoy, "Term proximity scoring for keyword-based retrieval systems*",In Advances in Information Retrieval*, LNCS, 2003.

[19] O. D. Kretser, A. Moffat, "Effective document presentation with a locality-based similarity heuristic*",*in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999.

[20] D. Petkova, W. B. Croft, "Proximity-based document representation for named entity retrieval" in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management,* 2007.

[21] Y. Lv, C. Zhai, "Positional language models for information retrieval", in*Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009.

[22] C. Zhai, "Statistical Language Models for Information Retrieval A Critical Review",*Found. Trends Inf. Retr., 2*(3), 137-213, 2008.

[23] D. Hiemstra, "A Linguistically Motivated Probabilistic Model of Information Retrieval", in *Proceedings of the second Eurapean conference on Research and Advanced Technology for Digital Libraries*, pp. 569-584, Springer-VerlagLondon, UK, 1998.

[24] D. R. H. Miller, T. Leek, R. M. Schwartz, "A hidden Markov model information retrieval system". In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999.

[25] X. Liu, W. B. Croft, "Statistical language modeling for information retrieval",*Annual Review of Information Science and Technology, 39*(1), 1-31, 2005.

[26] W. B. Croft, D. Metzler,T. Strohman, "Search engine : Information retrieval in practice", Retrieved 1/5/2012, from http://www.search-engines-book.com/collections/, 2012.

[27] Lemur.Project. About The Lemur Project. Retrieved 1/3/2012, 2012, from http://www.lemurproject.org/about.php, 2010.

[28] I. Red Hat, Cygwin. Retrieved 1/3/2012, 2012, from http://www.cygwin.com/, 2012.

[29] NIST. Text REtrieval Conference (TREC). Retrieved 20/1/2012, 2012, from http://trec.nist.gov/, 2012.

BIOGRAPHY

Huda Mohammed Barakat is a student in University of Malaya (UM), faculty of Computer Science and Information Technology. She is attached to the Department of Information Systems and she has just completed her Masters studies in the field of information retrieval.

MaizatulAkmarBinti Ismail holds a PhD from University of Malaya (UM). She is currently the head of Information System Department at faculty of Computer Science and Information Technology at UM. Her work deals with information retrieval, Semantic Web, Ontology and Integration of Heterogeneous Databases.

Sri Devi Ravana is a Senior Lecturer in Computer Science at the Faculty of Computer Science & Information Technology and the Head for Knowledge Engineering and Informatics Research Group at The University of Malaya, Malaysia. She holds PhD degree in Computer Science from The University of Melbourne, Australia. Her main research area is in the field of Information Retrieval and Data Engineering.