Measuring the Stability of Query Term Collocations and Using it in Document Ranking

by

Rana Alshaar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2008

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Delivering the right information to the user is fundamental in information retrieval system. Many traditional information retrieval models assume word independence and view a document as bag-of-words, however getting the right information requires a deep understanding of the content of the document and the relationships that exist between words in the text.

This study focuses on developing two new document ranking techniques, which are based on a lexical cohesive relationship of collocation. Collocation relationship is a semantic relationship that exists between words that co-occur in the same lexical environment. Two types of collocation relationship have been considered; collocation in the same grammatical structure (such as a sentence), and collocation in the same semantic structure where query terms occur in different sentences but they co-occur with the same words.

In the first technique, we only considered the first type of collocation to calculate the document score; where the positional frequency of query terms co-occurrence have been used to identify collocation relationship between query terms and calculating query term's weight.

In the second technique, both types of collocation have been considered; where the co-occurrence frequency distribution within a predefined window has been used to determine query terms collocations and computing query term's weight. Evaluation of the proposed techniques show performance gain in some of the collocations over the chosen baseline runs.

Acknowledgements

First and foremost I would like to thank my supervisor, Dr. Olga Vechtomova, for her support and encouragement. I deeply appreciate her valuable advice and guidance during the course of this research. I also want to thank my thesis committee members, Dr. Charlie Clarke and Dr. Mark D. Smucker, who provided me with valuable and constructive comments to improve this thesis.

I would also like to acknowledge and express my deepest appreciation to my parents, Omar and Husen, my sisters Randa, Maissaa, Rima and my brother Hisham, for their support and love.

Most importantly, I would like to thank my husband Osama, who encouraged me to pursue a master degree and who always believed in me and provided me with a great advice when I needed. His love and support allowed me to balance my time between my family and my study. Finally, I would like to mention my son Amir, who was around me all the time with a wide smile, which helped me to overcome all the challenges.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

Before the introduction of the World Wide Web in 1990s, the use of information retrieval systems was limited only to librarians and information professionals who are in charge of classifying and organizing documents in library databases, and performing the search tasks on behalf of the end users. The responsibility of these specialists was to understand the user's need and then interpret this need in a way that will extort the right information to the user.

With the introduction of the World Wide Web and the rapid development in communication technology, and the huge number of users who are looking for relevant information to satisfy their needs, search task has been shifted from the librarian and information professional to the actual user, who may have little or no knowledge about the structure of the implemented system. Although these advances have been seen as a huge success, the huge number of documents that are available to a broad range of untrained users make the search task a complex and tedious task.

This complexity and difficulty of getting the right information created the urgency to provide a high scalable information system that responds efficiently and effectively to the user's request. In order to meet this need, many researchers focused their attention in the area of information retrieval, each with particular interest. Some researchers are more interested in the representation of the documents and the user query, while others have an interest in developing a better

documents matching and ranking techniques. In this study, we follow the second interest by exploring the use of collocation and term proximity to develop a new document ranking method.

1.2 A definition of Information Retrieval

Information retrieval (IR) is about searching documents for information that meet a user need. It is concerned with the representation, storage, organization of, and access to information items that make retrieving information an easy and beneficial task (Baeza-Yates, Ribeiro-Neto 1999). IR doesn't deal with the documents as a plain text, but rather a representation of each document is formulated automatically or manually. Traditionally, documents' representations are formulated by extracting meaningful words (index terms) from these documents and indexing them. This set of keywords provides a logical view of the documents. When the user sends a request, a representation of his request will also be formulated in the same manner. Then the user query (request representation) and the representation of the document will be matched according to specific matching algorithm; then results are presented to the user in a form of a ranked list that contains the most relevant documents at the top of the list. Most of the time the documents that are delivered to the user are irrelevant because of the way those indexes are being matched. This problem raises an important issue of deciding which documents are relevant and which are not. As a result, different information retrieval models are developed for this purpose.

An information retrieval model forms the base that each particular information retrieval system is built on. For example, SMART retrieval system is based on the Vector Space Model (Salton, 1971), while Okapi retrieval system is based on Robertson and Spark Jones' Probabilistic Model (Spärck Jones et al., 1998). In deciding which IR model to adopt, we have to distinguish between

two user's tasks, ad hoc and filtering retrieval task. In ad hoc, the number of documents in the collection remains constant and the queries that are sent are the ones that vary. Filtering task on the other hand is different in that the queries are constant but the number of documents is changing as new documents are always arriving to the system. Some models are suitable for both tasks such as the classic models and many other models will not be mentioned here. In this study, we will use the Okapi retrieval system and we will only focus on ad hoc retrieval task, to develop new methods of document ranking.

1.3 Overview

A document ranking technique is an algorithm that tries to match documents in the corpus to the user query, and then ranks the retrieved documents by listing the most relevant documents to the user query at the top of the ranking. There are many algorithms that have been developed for this purpose, each with its own mechanism. Some techniques treated the query as a set of independent terms and they concentrated on finding documents that include these individual terms, while other techniques treated the query as one complete phrase and documents that contain the whole phrase will be ranked higher in the list. Although these techniques have shown some improvements, still this improvement was not consistent across different queries. Terms in the query may be related to each other differently, some terms are strongly related to each other while others not. In this work, we explore the use of one association measure to determine the query terms that form a collocation and the ones that don't and then we investigate the use of term collocation information and proximity in document ranking.

The thesis is organized as follows: chapter 2 provides background information about the classical models of information retrieval. Chapter 3 contains an overview about the two concepts of lexical cohesion and collocation. It also contains a summary of different methods of collocation extraction. Chapter 4 presents information about the importance of using term proximity and the different document ranking methods that incorporate term proximity. Chapter 5 describes our new document ranking methods using collocation. Chapter 6 includes performance evaluation and discussion of the analysis results. Chapter 7 presents a conclusion and future work.

2. Classic Models of IR System

An IR model can be characterized by four elements: (1) representations of the documents, (2) a representation of the user's query, (3) matching strategies for assessing if the document is relevant to the query or not, (4) and finally a method for ranking the results of the search.

The classic models are the Boolean Model, the Vector Space Model, and the Probabilistic Model. All these models consider index terms as a representation of a document, which is a single or a group of words that are usually nouns or noun phrases. The number of index terms from these documents is vast and usually not all of them are useful for describing these documents. As a result, there should be a way of assigning importance to each index term. The concept of weight has been introduced to indicate how important the term in describing the subject of the document. Each weight is associated with the term and the document where that term appears.

2.1 Boolean Model

Boolean model is one of the first models in information retrieval. As the name indicates, this model is based on Boolean logic where the user's query and document representations consist of a set of index terms. The model represents the user's query as a Boolean expression using Boolean operators: AND, OR, and NOT. For instance, documents retrieved for the query that contains two terms "collocation" and "extraction" will be different depending on the logical operator that is used to combine the two words. If the two terms are combined using AND ("collocation" AND "Extraction"), such query will only match documents that contain the two

words. On the other hand, if OR operator is used ("collocation" OR "extraction"), then any document that contains either one of the two words will be retrieved. In figure 2.1, circle 1 represents all the documents that contain the term "collocation", while circle 2 represents all the documents that contain the term "extraction". The highlighted section in figure 2.1 A includes the documents retrieved set for the query with AND operator, and the highlighted section in figure 2.1 B represents the documents retrieved set for the query with OR operator.

Figure 2.1 Boolean Model

To find a document that match a user query, terms are assigned weights that are binary, one if the term is present in the document, no matter how often it occurs in a document and zero if the term is absent from a document. Boolean expression is calculated using these weights and then only documents that exactly match this expression (have a score of one) will be retrieved without any ranking consideration.

Although the Boolean model is easy to implement, it has two limitations that are associated with it. Firstly, formulating the query as Boolean expression for inexperienced user is usually impractical. Nevertheless, experienced users are in favor of using Boolean search since it returns documents that exactly match their request (Cleverdon, 1984). Secondly, by using binary weight only documents that exactly match the user query will be returned. Requiring this exact matching is more as a form of data retrieval instead of information retrieval since there might be a lot of

documents that are of interest to the user but are not retrieved, because they don't exactly match the query expression.

2.2 Vector Space Model

In the Vector Space model, the document and the query are represented by vectors of index terms in a multi-dimensional Euclidean space, where each index term in the corpus is associated with one dimension. The cosine measure of similarity is used to determine the similarity between a document and a query. This measure will determine the cosine value of the angle between a document and a query vectors. If the two vectors are perpendicular, then the value of the cosine is zero. If the two vectors are parallel, then the value of the cosine is one. Otherwise, the value will be between one and zero. This similarity value is calculated based on the following equation (Eq.2.1):

$$S(D,Q) = \frac{\sum_{t \in Q} w_{t,D} \times w_{t,Q}}{\sqrt{\sum_{t \in Q} w_{t,D}^2 \times \sum_{t \in Q} w_{t \in Q}^2}} \qquad (2.1)$$

$w_{t,D}$ is the term's weight in document D.

$w_{t,Q}$ is the term's weight in the query.

The term weight is influenced by two factors, term frequency and inverse document frequency. Term frequency "*tf*" has been defined as the number of times the term appears in a document. It is valuable because the term that appears more frequently in the document is a better candidate for describing this document. On the other hand, inverse document frequency "*idf*" refers to the number of documents in the collection that contain a particular term. According to this measure, as the term appears in the majority of the documents, it is no longer a predictor of relevance and

this will cause its weight to fall. The term's weight is determined according to the following formula (Eq. 2.2):

$$w_{t,D} = \frac{tf_{t,D} + 1}{maxtf_D + 1} \times idf_t \quad (2.2)$$

$maxtf_D$ is the maximum frequency a term has in document D.

By using non-binary weights, Vector Space model provides a better performance than the Boolean model. First, it allows documents that match the user query partially to be retrieved, which is beneficial in retrieving documents that contain information of interest to the user, even if it doesn't exactly match the user's query. Second, it provides a way of ranking retrieved documents based on their similarity to the query, which frees the user of going through a tedious task of checking many documents to find relevant information, as in the Boolean model.

The main disadvantage of the Vector Space model is index term independency, where a weight is assigned to each index term without taking into account the presence or absence of any other terms in the document. By not including term dependency into the model, the model will not distinguish between a document that contains query terms in close proximity to one another and a document that contains query terms in different sentences. While in practice, the user may be more interested in the first document than the second one and therefore a term's weight should be determined based on the occurrence of other terms in the document.

2.3 Probabilistic Model

One of the well established and commonly used models in IR is the Robertson and Spärck Jones probabilistic model (Spärck Jones et al., 1998). This model goes contrary to the Boolean model,

which tries to find documents that exactly match the user query, by focusing on finding documents that partially match the user's need. It also focuses on finding documents that match the user's need rather than the query; because the formulation of the query may not describe exactly what the user is looking for.

They proposed their first probabilistic model in 1976 that is also known as binary independence retrieval (BIR) model, because it uses binary weight for index terms and assumes independency between these terms. Based on the probability theory to estimate the probability that document d_j is relevant to the user query, it ranks documents in decreasing order of their probability of relevance and present them to the user. This ranking process determines system effectiveness, since documents that are estimated by the system as the most likely to be relevant are presented to the user first. From this point of view, the goal of the probabilistic model is not just to retrieve relevant documents, but to rank these documents based on their probability of relevance to the user. It depends on the idea that terms are independent and the score of each term is calculated without considering the presence of other terms in the same context, regardless of the distance between their occurrences and without any consideration to any relationship that might exist between query terms in a document

For each document in the collection, the model estimates what is the probability that this document is relevant to the user or not. To make this decision, the model must consider the representation of such document. Since documents and queries are represented by index terms, the calculation of the probability of relevance is extended to include these terms. According to this, the model needs to estimate what is the probability that each index term will be represented

in a document that is relevant. To see how these index terms contribute to the relevance of a document, each index term is assigned a weight based on its presence and absence in a document (Eq. 2.3).

$$w_i = log \frac{p_i(1-\overline{p}_i)}{\overline{p}_i(1-p_i)} \tag{2.3}$$

p_i is the probability that term t_i is present in relevant documents.

$\overline{p}_i$ is the probability that term t_i is present in non-relevant documents.

After calculating the weight for each term, each document is given a score, which is calculated by adding the weights of all the terms that belong to it. Documents with the highest score will be presented at the top of the list as the most relevant documents.

In order to determine the term's probability of presence or absence in relevant documents, Robertson and Spärck Jones realized that a query term that appears in few documents is a better predictor of relevance than those terms that appear in most of the documents. According to this, a new weighting function has been introduced (Eq. 2.4)

$$CFW = \log \frac{N}{n_i} \tag{2.4}$$

N is the number of documents in the collection.

n_i is the number of documents that contain the term i.

They also realized that relevance information that might be available would provide a better way for estimating term's weight. Therefore, they introduced the Term Incidence Contingency Table, shown in Table 2.1.

	Relevant	**Non-Relevant**	
Containing the term	r	N - r	n
Not containing the term	R - r	N - n - R + r	N-n
	R	N - R	N

Table 2.1 Term Incidence Contingency Table

Considering relevance information in table (2.1), a new term's weight formula has been introduced (Eq. 2.5):

$$W_i = log \frac{(r + 0.5)(N - n - R + r + 0.5)}{(R - r + 0.5)(n - r + 0.5)} \quad (2.5)$$

R is the number of documents that are known to be relevant.

r is the number of the documents that are relevant and contain the term i.

N is the number of documents in the collection.

n is the number of documents that contain the term i.

As a result, terms that occur in most of the relevant documents will have a higher weight than terms that occur in few relevant documents.

In the above equation 0.5 is added to each cell value to avoid the problem of having incomplete relevance information that may lead to infinite weight.

Relevance information is an important factor for determining the probability of relevance, but getting this information is crucial. One way of getting this information is by using the user feedback on retrieved documents, who indicates documents that are relevant and the ones that are not. The other way is by using blind relevance feedback, where a set of documents will be

retrieved and then the top k retrieved documents will be considered as the relevant documents set; then based on this information a new search will be initiated.

Robertson and Spärck Jones (1998) realized that taking into account only the presence of the term in a document to calculate document score is not sufficient, because even if the term is present in a document, it may not be related to the main subject of the document; therefore, it may not be helpful to determine the relevance of the document. In order to predict more accurately if the index term is related to the document's subject or not, the number of times the term occurs in a given document should be taken into account. According to this, an index term will have a higher weight if it occurs in few documents in the collection and occurs more frequently in the given document.

By including term frequency in the weighting function, a new consideration must be taken into account. Term frequency must be considered in relation to the document length. Although a term may have the same frequency in two documents with different lengths, it should be given a higher weight in relation to the short document since it describes its content better than the long one. Thus the term weighting function is further modified to include this information (Eq. 2.6).

$$BM25_{w_t} = W_i \times \frac{(k_1 + 1)tf_{t,D}}{k_1\left((1 - b) + b \times \frac{dl}{avdl}\right) + tf_{t,D}} \tag{2.6}$$

Where k_1 is a normalization parameter of term frequency. Its default value is empirically set to 1.2. It has been introduced because Robertson and Spärck Jones (1998) realized that the optimal performance was achieved when document score is not linearly dependent on tf.

b is a normalization parameter of document length. Its default value is empirically set to 75.

W_i is the term's weight that can be calculated according to equation (2.5) if relevance information is available, otherwise it will be calculated according to equation (2.4).

Robertson and Spärck Jones (1998) realized the inaccuracy and the simplicity of the term independency assumption, so they did a further study in order to include term dependency in their model by using phrases, but the experiment results only showed a small improvement.

3. Cohesion and Collocation

When reading a text, we as humans don't see the text as a series of words, but rather unintentionally we try to link all these words in order to have a full understanding of what the text is all about. To make such understanding possible, two properties of the text, coherence and cohesion, must be present. Coherence is a property of a text where each part of the text adds to the total meaning, so the reader can follow through the text smoothly. Cohesion on the other hand is more related to how elements of the text are structured grammatically and semantically. Halliday and Hassan (1976) indicate that cohesion is very important in preserving the unity of the text and in determining its main subject. In the following example, the two sentences are related together, where the word "them" in the second sentence refers to the word "ingredients" in the first sentence. This reference relation lets the two sentences form a cohesive relation with each other, and allows the user to understand the second sentence by relating it to the first one:

> Mix all the ingredients together until they blend together.
>
> Then put them in a baking sheet and bake for 15 minutes.

According to Halliday and Hassan (1976), a text is not made up of unrelated elements but rather it is a semantic entity that is achieved through the semantic relations that exist between its different parts. They refer to any semantic relation that exists between text's elements as a cohesive tie. They classify these relations into five categories: reference, substitution, conjunction, ellipsis and lexical cohesion. The first four categories are related to the grammatical structure of the text, while the last category is related to how terms in the text are semantically structured.

Reference: is a relation where the pronoun will be used in order to refer to a text unit that has been identified in the text preceding or following it. In the following example, the pronoun “she” refers to the name of the girl “Sally”:

Sally didn’t come to school yesterday. She was sick.

Substitution: is a relation where the word or a phrase is substituted by another word that is more general. In the following example, the word “one” is used to replace the word “dress”:

Which dress would you like to buy? I would like the red one.

Ellipsis: is a relation in which a word is eliminated after it has been mentioned in the previous context. In the following example, the word “I ordered” is omitted in the second part of the sentence:

What did you order? fries.

Conjunction: is a relation where a connector is used in order to form a relation between two sentences or statements. In the following example, the word “but” is used as a connector:

I would like to go to the trip, but I am sick.

3.1 Lexical Cohesion

A meaningful text is not just a group of words that are grammatically correct, but also the meaning of these words must be related to the topic of the text. Usually, words have different meaning and determining the appropriate meaning in a particular text is highly related to the contextual environment surrounding it (the words that occur before it or after it in a sentence or a text). These semantic relations between words are what give the text its main characteristic of being lexically cohesive.

Lexical cohesion is a property of text that is attained through the connection between words that are semantically related (Morris and Hirst, 1991; Halliday and Hassan, 1976). The existence of such semantic relations in text helps to determine the main subject that is covered throughout the text. Two major types of lexical cohesion have been identified by Halliday and Hassan (1976): reiteration and collocation.

Reiteration: includes a wide range of relations that exist between two lexical items in text such as reference and repetition of the same word relation, superordinates relation, subordinates relation and synonyms relation.

- Reiteration by means of superordinate/subordinates: is a relation where a word is used that is more general than another word mentioned in the text. For example, the term vegetable refers to a wide range of other terms such as broccoli. On the other hand, broccoli is a subordinated of the general word vegetable:

  I like to eat a lot of vegetable.

  My favorite is broccoli.

- Reiteration by using repetition: is a relation where the same word is used again in the same context:

  My son started going to school.

  Fortunately, the school is not far away.

- Reiteration by using synonymy: is a relation that is formed by using two different expressions that share similar meaning. So, in the example below 'child' and 'kid' are two different words but both refer to the same meaning.

  When I was a child, I used to have a lot of toys.

  Being a kid is really a lot of fun.

Collocation: is a type of lexical cohesion in which a semantic relation is formed between a pair of words that co-occur more often within the same context. These words could be in adjacent locations to each other or they could occur at a distance from each other.

3.1.1 Lexical Links, Bonds and Chain

Lexical link is a lexical cohesive relationship that exists between pair of words in text (Morris and Hirst, 1991). Hoey (1991) mentions that we don't perceive the text as separate cohesive links that exist between a pair of words in the sentences, but rather we perceive the text as set of sentences that are related to each other as one complete element of the text. Without these relations, the text will consist of separate sentences that don't share any topical meaning and the only thing that is common between them is the contextual environment that they occur in (contextual environment is the text that include these sentences). This cohesive relation between sentences is what Hoey (1991) refers to as a lexical bond. For this relation to exist a specific number of lexical links must be formed between these two sentences and this number should be determined in relation to the length and the degree of lexical relationships that sentences of the text have.

Lexical cohesion in text is generally recognized through a series of words spanning throughout the text and that form lexical links with each other – lexical chains (Morris and Hirst, 1991). Halliday and Hassan (1976) define the concept "chain" as a relation where a term has a relation with a previous term, which in turn has a relation with another previous term and so on.

3.2 Collocation

Although collocation is an important and common phenomenon, different researchers define collocation differently. Benson (1990) gives the following definition for collocation "collocation is an arbitrary and recurrent word combination". Manning and Schütze (1999) define collocation as "A collocation is an expression consisting of two or more words that correspond to some conventional way of saying things". Firth (1957) introduces one of the earliest definitions of collocation "Meaning by collocation is an abstraction at the syntagmatic level and is not directly concerned with the conceptual or idea approach to the meaning of the words". Choueka (1988) provides a definition that is similar to Firth's definition in focusing on the linguistic characteristic of collocation: A collocation is defined as "a sequence of two or more consecutive words, that have characteristics of a syntactic and semantic unit, and whose exact and unambiguous meaning or connotation can't be derived directly from the meaning or connotation of its components". Choueka's definition is restrictive in that it assumes that collocate words are adjacent to one another, while in general collocate words could be separated by other words. Morris and Hirst (1991) define collocation as a relation that exists between words that occur in the same lexical environment.

3.2.1 Collocation Characteristics

Manning and Schütze (1999) recognize the following characteristic of collocation:

- Non-compositionality: the meaning of a collocation can't be recognized from the meaning of its constituent words. For example, 'strong tea' as a collocation has a meaning that is different than the meaning of the two words 'strong' and 'tea'.

- Non-substitutability: the non-compositional property of collocation makes it impossible to substitute any of the consistent word with its synonym, because as the word is changed, the meaning of the collocation will change too. For example, if the word "United" is replaced by "Joint" in "United Nations", the total meaning of the expression will change.
- Non-modifiability: it is not possible to modify the individual words in collocation by changing their order or by adding more words to the expression. For instance, "Once in a blue moon" can't be modified to "in a blue moon once".

3.2.2 Collocation Properties

Smadja (1993) recognizes four properties of collocations that have an important application in computational linguistics:

- Collocations are arbitrary: since the total meaning of collocation is different than the meaning of its constituent words, it is difficult for a person whose first language is not English to form or identify collocations in text by translating them word-for-word.
- Collocations are Domain-Dependent: collocations are used and understood differently based on their domain of use. Technical terms and jargons are used differently by technical people, who are familiar with them, compared to non-technical people, who are not familiar with the right meaning of these terms.
- Collocations are Recurrent: the co-occurrence of collocation words is not occasional, but rather these word combinations co-occur frequently in a particular lexical environment.
- Collocations are Cohesive Lexical Clusters: Smadja refers to cohesive cluster as: "The presence of one or several words of the collocations often implies or suggests the rest of

the collocation". This gives an indication that collocation's words have a strong tie of co-occurring with one another. According to computational linguistics, this means that the probability of occurrence of all words together is larger than the multiplication of the probability of the occurrence of each individual word.

3.2.3 Types of Collocation

Collocation expressions differ greatly based on different factors such as the number of collocate words, the type of syntactical group that the words belong to, the type of syntactic relation that the words are engaged in, and the strength of the co-occurrence tie between these words (Smadja, 1993). According to this, different types of collocation have been identified by Smadja:

Predicative Relation: predicative relation is composed of two words that usually occur together in a comparable syntactic relationship. These words have a great flexibility of appearance, in terms of the number of words that occur between them and the order in which they appear. This flexibility of appearance makes it harder to identify them. For instance the two words "make" and "decision" could be adjacent to one another "make decision" or they might be separated by other words "make an important decision".

Rigid Noun Phrases: as the name indicates, this type of collocation includes words that are engaged in an unbending relationship where the composed words always appear in the same sequence and any changes to their structure, such as adding or removing words, will result in altering their meaning. An example of a rigid noun phrase is "interest rate".

Phrasal Templates: phrasal templates correspond to word phrases, which may contain one or more empty slot or they may not contain any slot. Empty Slot is filled by a word that has a particular part of speech that is indicated by the phrasal templates.

3.3 Collocation Extraction

Recognizing the importance of collocation, many researchers focused their attention on developing a new approach to the automatic extraction of collocations from a corpus. Although some of the developed approaches showed good results of retrieving important collocations, each one of these approaches has its limitation. In this section a brief description of these approaches is discussed. A more complete description is provided by Manning and Schütze (1999).

One of the initial methods of retrieving collocations is the one proposed by Choueka et al (1983), which is based on the number of times collocate words appear together in the corpus. Choueka et al. understand collocation as a set of adjacent words that have a high tendency to recurrently appear together. Although they realize that these collocations could be of variable length, they only focus on a limited set which includes two to six words. In their methodology, they use frequency to retrieve this kind of collocation. Only collocations, which have a frequency higher than a predefined threshold, will be retrieved accordingly. They performed their test on an 11 million-word corpus from the New York Times archives. Their experiments were effective in retrieving thousands of collocation. Despite this effectiveness and the simplicity of implementation, the frequency approach has its drawback, where it is only suitable for fixed phrases such as "United Nations", where the words exhibit a rigid way of appearing together.

Mutual Information

Mutual information (MI) is a concept in information theory that was originally introduced by (Fano, 1961). Mutual information score is used for identifying interesting co-occurrences of terms. This score "compares the probability that the two words are used as a joint event with the

probability that they occur individually, and that their co-occurrences are simply a result of chance" (McEnery 1996, p. 71). The standard formula for expressing mutual information score between pair of words is:

$$I(x,y) = log_2 \frac{P(x,y)}{P(x)P(y)} \quad (3.1)$$

Where $P(x,y)$ is the joint probability of occurrence of the words x, y; $P(x)$ and $P(y)$ are the individual probabilities of occurrence.

(Church et al. 1991, 1989) used mutual information to identify collocations from text. In their implementation, they determine the probability of term occurrence based on the relative frequency; where *P(x)* and *P(y)* are the number of times each word appears in the corpus normalized by the size of the corpus; *P(x, y)* is the number of times the two words occur together in a fixed-size window (usually 5 words) normalized by the size of the corpus. Although this measure has an advantage of being able to extract distant word collocations, it has a limitation of being in favor of low frequency words. In other words, it rewards words with low frequencies more than words with high frequencies. To overcome this limitation, a frequency threshold of at least 3 could be specified, so only words with a frequency higher than this threshold will be considered or by multiplying the mutual information score by the joint frequency (Manning and Schütze, 1999).

The t-test

The t-test is a statistical association measure that looks at the mean and variance of a sample of measurements. The test assesses the difference between the expected means ($\overline{x}$) and the observed

means of a normally distributed data (μ), normalized by the variance of the sampled data (s^2), which in turn is normalized by the size of the data (N).

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}} \tag{3.2}$$

The t-test has been criticized for assuming normal distribution for words being sampled, which is not always true for natural language data. Nevertheless, t-test has been seen as a useful measure for ranking collocations.

Pearson's chi-square test

The Pearson's chi-square test (X^2 test) is another statistical approach for collocation extraction. While t-test looks at the difference between the expected and observed means, X^2 test indicates the degree of association between words by comparing the observed and expected frequencies, and based on the difference between the two frequencies the null hypotheses of independence will be rejected or accepted.

Contrary to the t-test, X^2 test does not assume that terms are normally distributed, which makes it more appropriate for the extraction of word collocations. But this is only held true when the sample size is large enough, because X^2 test tends to overvalue a data when the data is sparse.

Likelihood Ratio

Likelihood ratio test is used for collocation extraction by comparing two hypotheses to determine which hypothesis is more probable to occur more than the other. The first hypothesis proposes

that two terms occur independently from each other, while the second hypothesis proposes that the occurrence of one of the terms is dependent on the occurrence of the other term.

Whereas mutual information and X^2 test have been criticized for not being appropriate for sparse data, likelihood ratio overcomes this limitation by being useful when the data is sparse. In addition, it is easier to interpret likelihood test score than the X^2 test score.

3.3.1 Xtract System

Smadja (1993) realized that a collocation doesn't have to be a fixed phrase where the two words have to be adjacent to one another or always occur at the same distance from each other. Two or more words form a collocation even if there are intervening words between them, as long as their co-occurrence pattern is frequent enough. According to this, Smadja implements a set of techniques to retrieve collocations based on statistical measures and syntactical information. These techniques are integrated to form a well known lexicographic tool "Xtract" that focuses on collocation identification and extraction from text based on words co-occurrence statistics.

Smadja's technique for collocation extraction is better than previous extraction techniques in two ways: first, it focuses on extracting both contiguous and non-contiguous word collocations. Second, n-gram collocations (collocations that contain n words) are extracted along with bi-gram collocations (collocations that contain only two words).

Collocation extraction using Xtract involves three main stages. In what follows, I will give a brief description of these stages and a more detailed description of the first stage, which we used

in our study to determine stable bigrams in the query, will be described in the methodology section.

1. The first stage of Xtact is concerned with extracting important bi-gram collocations based on statistical analysis, where only bi-grams that occur together more frequently and exhibit a rigid way of appearance will be extracted.
2. In the second stage, n-gram collocations are constructed based on the bi-grams obtained from the first stage.
3. The third stage is different than the previous two stages by using syntactical information in order to eliminate any bi-grams that are insignificant.

3.4 Using Collocation in Document Ranking

The importance of collocation has been recognized in many natural language applications such as query expansion (Vechtomova et al., 2003), term weighing (Hisamitsu and Niwa, 2002), and topic segmentation (Ferret, 2002). Only recently, a study by Vechtomova et al. (2008) showed the usefulness of collocation in document ranking. Vechtomova et al. (2008) developed two document ranking methods by using lexical cohesive relationship between query terms. The first method depends on using collocation relationship between query terms in the same sentence to predict if the document is relevant or not. This method is described in section (4.3). The second method (*lexical bonds* method) depends on using collocation relationship between query terms, assuming that query terms appear in different sentences and a same word appears in these two sentences. In this method, the document matching score is calculated as the sum of all query terms' weight, where the term's weight depends not on the term frequency but rather on the pseudo frequency, which is the sum of the contribution value of every instance of a query term.

The contribution value for each instance is based on the number of lexical bonds the sentence that contains this instance has with other sentences in the documents (Eq. 3.3).

$$c(t_i) = 1 + n \times \frac{Bonds(s)}{AveBonds} \tag{3.3}$$

Where: $Bonds(s)$ is the number of bonds sentence s forms with other sentences in the document; n is a normalization factor that has a value between zero and one; $AveBonds$ is the average number of bonds in the document, it is determined as the total number of bonds all the sentences in the document have with each other normalized by the total number of sentences in the document.

The experiment results indicated that the new document ranking method performed better than the BM25 and BM25tp document ranking functions that are implemented in Wumpus IR system (Büttcher et al., 2006). However, this improvement was not consistent across different document collections and not consistent across different topics within the same document collection.

4. Term Proximity

In this work, we explore the use of term proximity in document ranking. Term proximity is used to refer to the distance between pair of query terms that form a collocation. There are many studies that explored the use of term proximity in document ranking. Each of this method has its limitations, which we will discuss later in this section.

4.1 Document Retrieval and Ranking

The main focus in the area of information retrieval is satisfying the user need by returning the most relevant information. One of the most important aspects in IR that captures the focus of many researchers is providing a highly efficient and effective retrieval technique, which retrieves the most relevant documents and rank them at the top of the list. One of the earliest and effective techniques to retrieve and rank documents in IR was based on term frequency (Salton, 1971; Spärck Jones et al., 1998). Using term frequency to determine the relevance of the document was the focus of many traditional information retrieval models and it goes back as early as 1958, Luhn asserted (Luhn, 1958):

"it is here proposed that the frequency of word occurrences in an article furnishes a useful measurement of word significant. It is further proposed that the relative position within a sentence of words having gives values of significance furnishes a useful measurement for determining the significance of sentences. The significance factor of a sentence will therefore be based on a combination of these two measurements."

Luhn's statement emphasizes two important aspects to determine relevance:

1. The frequency of term occurrence, which was the focus of most of the approaches in information retrieval. The two most commonly used frequencies are term frequency and inverse document frequency. Term frequency "*tf*" is defined as the number of times the term appears in a document. Inverse document frequency "*idf*" refers to the number of documents in the collection containing a particular term. The most popular models that are based on *tf*idf* to determine document's relevance are Robertson and Spärck Jones' probabilistic Model (Spärck Jones et al., 1998), and the Vector Space Model (Salton, 1971). These models view the document as *bag-of-words* without any consideration to the underlying semantical and syntactical structure or term proximity in the text.
2. The position of the term in a document, which attracted the interest of many researchers who realized viewing query terms as mutually independent is limiting, and more thoughts should be given to the position of query terms in a document to determine its relevance. From this point of view, some of these researchers focused their attention on developing new document ranking techniques, which depend on using proximity measures between terms' positions to determine document's relevance (Clarke et al., 2000; Büttcher et al., 2006; Vechtomova et al., 2008). All these measures are based on the distance between distinct term instances. Hawking et al. (1996) mentioned that the meaning of term distance is used differently in the area of information retrieval, but many of the proposed term proximity ranking functions are concerned with lexical distance, which is expressed as the number of words separating the occurrence of two query terms. In our study, we also use the term *distance* to refer to the lexical distance between query terms. While others focused on using phrases in document ranking (Fagan 1989; Mitra et al., 1997),

Mitra et al. (1997) examined the effectiveness of using statistical and syntactical phrases in IR at high-precision level. They defined a statistical phrase as a pair of adjacent non-stopwords that occur in at least 25 documents. Syntactical phrase has been defined as a set of words whose Part-of-Speech sequence follows a predefined syntactical structure (e.g. Noun-Noun, Adjective-Noun). Their research showed that phrases only provide a small improvement when a poor performance is attained using single terms and there is no difference in effectiveness between using syntactical and statistical phrases. Fagan (1989) conducted a complete comparison between syntactical and statistical phrase indexing and their effectiveness in IR. The evaluation results showed that using phrases provided better performance than using single terms and that the performance of statistical phrases is indifferent from that of syntactical phrases. Metzler and Croft (2005) explored term proximity information within the language modeling framework by modeling single terms, ordered phrases, and unordered phrases via a Markov random field. Their model showed a significant improvement, especially on large collections.

4.2 The importance of using Term proximity

Although the traditional models were effective in retrieving relevant information by using term frequency, considering query terms as mutually independent doesn't take into account the following two facts:

1. Words have different meaning and a particular meaning can only be recognized and clarified by analyzing the context where the word occurs. For example, if the user is looking for documents that talk about "traffic jam", a document that contains the two words in close proximity is more likely to be about rush hour and traffic rather than about

food. So, the meaning of the word "jam" can be clarified by using proximity relationship with other terms. According to this, a document that contains query terms in close proximity will be ranked higher than other documents where the query terms appear apart from each other.

2. Usually different topics or subjects are discussed throughout a document; therefore, words that occur in the same contextual location and close to each will be more likely related to the subject that is discussed in that particular part of the text.

4.3 Term Proximity in Document Ranking

In this section we will present a number of document ranking techniques that go beyond the *bag-of-words* assumption and that focus on using term proximity information by adopting the following two assumptions:

1. The more query terms the document has, the more likely that the document will be relevant.
2. The closer the query terms appear in a document, the higher the relevance score the corresponding document will have.

Rasolofo and Savoy (2003) proposed a relevance score that integrates proximity information with the Okapi weighing score (Robertson and Spärck Jones, 1998). The idea behind their approach of combining term proximity score with the Okapi score is to increase the effectiveness of the retrieval model by showing improvement at the top ranks. According to their assumption, term's weight will decrease as the distance between two distinct term instances increases. Given an instance of query terms pair(t_i, t_j), the *term pair instance- tpi* weight is:

$$tpi(t_i, t_j) = \frac{1.0}{d(t_i, t_j)} \quad (4.1)$$

Where $d(t_i, t_j)$ is the distance between the two terms t_i, t_j, whereas only five words are allowed to occur between the two terms.

In each document, terms pair(t_i, t_j) may occurs more than one. Therefore, the total weight of (t_i, t_j) is obtained by summing all the *tpi* weights for this terms pair instances in the corresponding document. The total weight is presented in the following formula:

$$w(t_i, t_j) = (k_1 + 1).\frac{\sum_{occ} tp_i(t_i, t_j)}{K + \sum_{occ} tp_i(t_i, t_j)} \quad (4.2)$$

Where k_1 and K are the same constants as in the Okapi formula.

After determining the weight of all query terms pairs in a document, a document score will be the summation of the Okapi relevance score and the proximity relevance score:

$$RSV_{NEW}(d, q) = RSV_{okapi}(d, q) + TPRSV(d, q) \quad (4.3)$$

Where the document proximity relevance score value $TPRSV(d, q)$ is computed as follows:

$$TPRSV(d, q) = \sum_{(t_i, t_j) \in s} w_d(t_i, t_j).\min(qw_i, qw_j) \quad (4.4)$$

Where qw_i and qw_j are the weights of the query terms t_i and t_j, which are based on the query term frequency and the total number of document that contain the query term in the corpus.

They did their evaluation on TREC ad-hoc test collections, where a noticeable improvement was obtained for precision at 5, 10 and 20 documents, and a slight improvement was attained in

average precision. As a result, their method is more effective only when a few documents are retrieved, which seems beneficial for users who are interested in the top ranked documents.

Büttcher et al. (2006) proposed a new method that is similar to Rasolofo and Savoy (2003) method, in term of combining Okapi BM25 weighing score with term proximity score. As contrary to Rasolofo et al. implementation, Büttcher et al. implementation didn't include any restriction on the length of the span between terms pair.

For every query term (T_j) in a document, they computed an accumulator score based on the distance between the position of this query term and the position of another query term (T_k) that precede it in text. Formally,

$$acc(T_j) \coloneqq acc(T_i) + w_{T_k}.\left(dist(T_j + T_k)\right)^{-2} \tag{4.5}$$

$$acc(T_k) \coloneqq acc(T_k) + w_{T_j}.\left(dist(T_j + T_k)\right)^{-2}$$

Where w_{T_k} is the inverse document weight of term T_k.

According to this formula, not only the current query term's accumulator score is increased, also the score of the term that occurs before it will increase too. Therefore, the term proximity score will be affected by other query terms that precede and follow it in a document. After determining the accumulator score of all query terms in a document, the document relevance score is determined as follows:

$$Score_{BM25TP}(D) = Score_{BM25}(D) + \sum_{T \in Q} \min\{1, w_T\}.\frac{acc(T).(k_1 + 1)}{acc(T) + K} \tag{4.6}$$

Where k_1 and K are the same constants as in the Okapi BM25 formula.

They did their evaluation on TREC Terabyte track collections, which showed improvement in precision at 10 and 20. They showed that the effectiveness varied if the query is stemmed or unstemmed and that the effectiveness is higher for stemmed queries than for unstemmed queries. In addition, the results have showed that the performance is different with different collections and as the collection size increases, the effectiveness of term proximity also increases.

Clarke et al. (2000) proposed a new document ranking technique that incorporates term proximity and Cover Density. They defined a cover as the shortest lexical span of words containing instances of all query terms. Their assumptions are (1) the shorter the span that contains a group of query terms, the more likely the corresponding document is relevant, and (2) the more spans are in a document, the more likely that the document is relevant.

Before calculating documents' relevance score, all documents were preliminary ranked and grouped into subsets according to the coordination level. Coordination level has been defined as the number of different query terms contained in a text. Within the same document subset, each document is given a relevance score by summing the scores of all covers that are contained in this document. The document's score is calculated according to the following formula:

$$S(d) = \sum_{j=1}^{n} I(p_j, q_j) \tag{4.7}$$

Where $I(p_j, q_j)$ is the score of the cover j that begin at term t_p and end at term t_q.

The score of each cover is based on its length as presented in the following formula:

$$I(p,q) = \begin{cases} \frac{k}{q-p+1} & if\ q-p+1 > k \\ 1 & otherwise \end{cases} \quad (4.8)$$

Where q and p are the positions of the two terms t_p and t_q.

They did their evaluation on TREC data set, which showed the effectiveness of their method in retrieving relevant documents for queries that contain one to three terms.

Vechtomova et al. (2008) also explored the use of term proximity in document ranking. In their approach, the document score is not the summation of the BM25 document score and the new proximity score, as we have noticed in the previous approaches, but instead they modified the original BM25 term weighting formula by using pseudo-frequency (*pf*) instead of term frequency, which is the sum of the contribution value of every instance of a query term *t*. The contribution value of each instance of the query term $c(t_i)$ is based on the lexical span between the i^{th} instance of this query term and the closest distinct query term that co-occurs with it, as expressed in the following equation.

$$c(t_i) = \begin{cases} 1 + \frac{1}{span(t,q)^p} & if\ q \in s; q \neq t_i; q \in Q \\ 1 & otherwise \end{cases} \quad (4.9)$$

p is a tuning constant. Its default value is empirically set to 0.5.

The experiment results indicated that the new document ranking method performed better than the BM25 and BM25tp document ranking functions that are implemented in Wumpus IR system (Büttcher et al., 2006). However, this improvement was not consistent across different document collections and not consistent across different topics within the same document collection.

4.4 Using Term Proximity in Query Expansion

Term proximity information was also proven to be useful in query expansion. Query expansion as a technique of reformulating the user query by adding additional terms to the original query, has been seen as a way to improve retrieval performance.

Vechtomova et al. (2006) used a combination of mutual information (MI) and proximity information to rank query expansion terms. Query expansion terms were extracted from the entire documents and then ranked based on the mutual information score and the distance between the original query term and the expansion term. The ranking score was proportional to the distance between the two terms and it increases with the frequency.

The experiment's results indicated that using a combination of mutual information and co-occurrence distance to select expansion terms provided better performance than no expansion, and also better than using MI alone.

5. Methodology

5.1 Introduction

Van Rijisbergen stated "The purpose of an automatic retrieval strategy is to retrieve all the relevant documents at the same time retrieving as few of the non-relevant as possible" (1975, p. 4). To achieve this, early studies on document ranking focused on different aspects of the documents to determine relevance. Robertson and Spärck Jones (1998) assumed independency between query terms in a document and proposed a document ranking function accordingly. Mitra et al. (1997) focused on using statistical and syntactical phrases in document ranking; and more recent studies focused on using term proximity information by incorporating distance between query terms instances in document ranking function.

The focus of our work is to rank documents based on term proximity information by focusing on the co-occurrence relationships between distinct query terms instances in a document (also referred to as collocation). Morris and Hirst (1991) defined collocation as a relation that exists between words that occur in the same lexical environment. While the roles of collocation in IR have been identified in the literature, only few studies focused on their role in document ranking.

In the proposed methods we explore the use of two types of collocation:

- Collocation in the same grammatical structures. This type of collocation represents a lexical cohesive relationship between terms that co-occur within a short span, such as a sentence (Figure 5.2), (Figure 5.1). In Figure 5.2, "radio wave" is a collocation in sentence (1).

- Collocation in the same semantic structure. This type of collocation represents a lexical cohesive relationship between query terms that co-occur with other words, where query terms span over a long distance (Figure 5.1) (Figure 5.2). In Figure 5.2 and 5.1, the term "wave" in the first sentence and the term "brain" in the second sentence are related by transitive collocation, where the two terms co-occur with the same words "research" and "discover". Since the two sentences contain two terms that are related by transitive collocation, a lexical bond relationship is formed between the two sentences (Vechtomova et al., 2008).

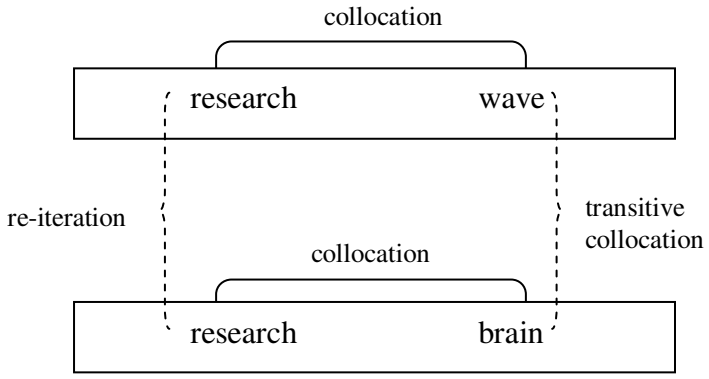

Figure 5.1 Example of the two types of collocation

Sentence 1: Many researches have been conducted to discover the side effect of radio wave on human health.

Sentence 2: One of the researchers discovered that using cell phone for a long time may cause brain cancer.

Figure 5.2 Sentences include collocation

Our assumption in this study is that query terms co-occurring in different part of the document, and the ones that co-occur in the same syntactical structure and have a strong relationship with each other could indicate the relevance of the document to the user's query. Therefore, we investigate the usage of these two types of collocation in document ranking by proposing the following two methods:

Method 1: document ranking based on query terms collocates that exist in the same sentence.

Method 2: document ranking based on query terms collocates that exist in the same sentence, and in different sentences.

5.2 Identifying Query Terms Collocation

A number of statistical association measures have been used for the identification of collocation in text, such as Pearson's chi-square test (X^2 test) and Log-likelihood ratio (Manning and Schütze, 1999), Mutual Information score (Church et al. 1991, 1989). A detailed description of these measures is given in section 3.3.

Natural Languages Processing (NLP) tools such as parsing and POS have also been used for the identification of collocations in documents, but they require a lot of time and computational resources. Therefore, in this study we decided to use statistical measures to find collocation.

As Halliday and Hassan (1976) mentioned, collocation is a type of a semantic relation that is formed between a pair of words that co-occur more often within the same context. These words could be in adjacent locations to each other or they could occur at a distance from each other. Following this definition, we will focus in this study on identifying and using both contiguous

and non-contiguous query term collocations that co-occur more frequently within the same context. To identify these collocates that stand in such flexible relationship, we use the same method that is introduced by Smadja and implemented in the Xtract system (Smadja, 1993). Our procedure for identifying collocation is as follows: All contiguous and non-contiguous query term pairs in the query will be identified and information about the co-occurrence frequency within a fixed size window (the size is set to 18 words, with 9 words to the right of the term t and 9 to its left) is obtained for each terms pair (t, t_i). This information includes their total frequency in the corpus within the predefined window, their frequency relative to the number of words that occur between them ($freq_1$, is (t, t_i) frequency when the two words occur adjacent to one another, and $freq_9$ is (t, t_i) frequency when the term t_i occurs 8 words apart from term t).

Based on this information, different levels of analysis are performed to determine query terms collocations. This analysis is introduced by Smadja and represented by the following set of inequalities (Eq. 5.1), (Eq. 5.2):

$$U_i = \frac{\sum_{j=1}^{10}(p_i^j - \overline{p_i})^2}{10} \geq U_0 \tag{5.1}$$

$$p_j^i \geq \overline{p_i} + (k_1 \times \sqrt{U_i}) \tag{5.2}$$

p_j^i is the frequency of the query term pair (t, t_j) where t_j occurs j words apart from t.

$\overline{p_i}$ is the average frequency of p_j^i.

U_0 and k_1 are a threshold that is set experimentally to (10, 1), as proposed by Smadja. In this study we will use the same value for U_0 and k_1. Experimenting with different values is left for future work.

The first inequality (Eq. 5.1) is used to determine which one of query terms pairs form a collocation, this condition will look at the co-occurrence frequency distribution of the two terms at different positions within the predefined window, where the collocate t_i appears at position j (-9 ≤ j ≤ 9) and then based on the histogram shape of this distribution, the interesting collocates will be determined. The shape of the p_j^i histogram is determined by the value of U_i. If U_i value is smaller than U_0 , then the histogram will have a flat shape and the frequency of the two terms are equally distributed in all the positions. If U_i value is bigger than U_0, then the histogram will have at least one peak where the frequency of the two terms in at least one of the positions is noticeably higher than the frequencies in any other positions. According to this, only query terms pairs that have a value of U_i greater than 10 will be considered as collocations.

After determining collocations that are formed between query term pairs, the interesting relative position (j) where the two terms co-occur will be identified by using the second inequality (Eq. 5.2), this condition helps in determining the important positions j (-9 ≤ j ≤ 9) where the relative frequency is above a predefined threshold.

5.3 Document Ranking using Collocates

The purpose of the work presented in this thesis is to explore the use of term proximity information (query term co-occurrence relationship) in document ranking. Term proximity information has attracted the attention of many researchers, where several studies have been conducted to explore the use of term proximity information in document ranking (Clarke et al., 2000; Büttcher et al., 2006; Vechtomova et al., 2008). Although our study depends on the same idea of using term proximity information, we explore this issue from a different perspective.

First, in most of these studies the value of the query term's weight is proportional to the distance between two query terms, so the term's weight will be higher if the term occur adjacent to another query term, and it will decrease as the distance between the two terms increases. Although in our research we focus on the distance, we try to determine how stable the co-occurrence of the two terms at a given distance, and then we modify the term's weight accordingly. Second, The major focus of the previous researchers were the distance factor, while in this study the frequency at particular distance is the main focus, in which we are interested in identifying query term pair collocate based on their positional frequency.

5.3.1 Method 1

After the identification of query terms collocates and the determination of the significant positions, where the terms pair has a frequency higher than a predefined value (as described in the previous section), a document matching score is calculated by using the original formula (Eq. 5.3).

$$MS = \sum_{t=1}^{|Q|} TW_t \tag{5.3}$$

Where $|Q|$ is the number of query terms.

TW_t is the weight of the term *t* that appears in the document.

The original formula of calculating the term's weight was introduced by Spärck Jones et al. (1998), which depend on the term frequency in the documents. Vechtomova et al. (2008) modified the term's weight by replacing the term frequency value with a pseudo-frequency (*pf*). In our implementation, we use Vechtomova et al (2008) implementation (Eq. 5.4).

$$TW_t = \frac{(k_1 + 1) \times pf_t}{k_1 \times NF + pf_t} \times idf_t \tag{5.4}$$

Where k_1 is the term frequency normalization factor.

NF is the document length normalization factor, which is calculated as follows (Eq. 5.5).

$$NF = (1 - b) + b \times \frac{DL}{AVDL} \tag{5.5}$$

In our approach, we follow the same principle of calculating the pseudo-frequency weights as in Vechtomova et al. (2008) method, where pf_t is calculated as the sum of the contribution values of every instance of the query term t (Eq. 5.6).

$$pf_t = \sum_{i=1}^{N} c(t_i) \tag{5.6}$$

Where *N* is the number of instances of query term *t* in the corresponding document.

Our calculation of the contribution value is as follows: for each query term instance, we find if there exists an instance of another query term that occurs at most 9 words apart on either side, and then based on their positional frequency at that particular distance we determine their contribution value (Eq. 5.7)

$$c(t_i) = \begin{cases} 1 + C & if\ FreqPercentage \leq 0.5,\ p_j^i \geq \overline{p_\iota} + (k_1 \times \sqrt{U_i}) \\ 1 & otherwise \end{cases} \tag{5.7}$$

Where *FreqPercentage* is the percentage of the frequency where the two terms occur *j* words apart (p_j^i) in relation to the highest frequency lp within the predefined window (Eq. 5.8).

$$FreqPercentage = \frac{lp - p_j^i}{lp} \tag{5.8}$$

5.3.2 Method 2

The second proposed method expands the bond method introduced by Vechtomova et al. (2008) from a different point of view. In their paper, the weight of the query term in a sentence is only affected by the number of bonds a sentence has with other sentences. While in our proposed method we realized that even if two sentences have the same number of bonds with other sentences in the document, the number of query term instances in these two sentences is not the same. Therefore, term's weight should be calculated by taking into account other query terms that co-occur with it in the same syntactical structure (such as a sentence), and whether if these terms form a stable collocation or not (Figure. 5.3). This figure shows that the number of Bonds each sentence s has with other sentences is the same, which is six, but each one of these sentences contains different number of query terms.

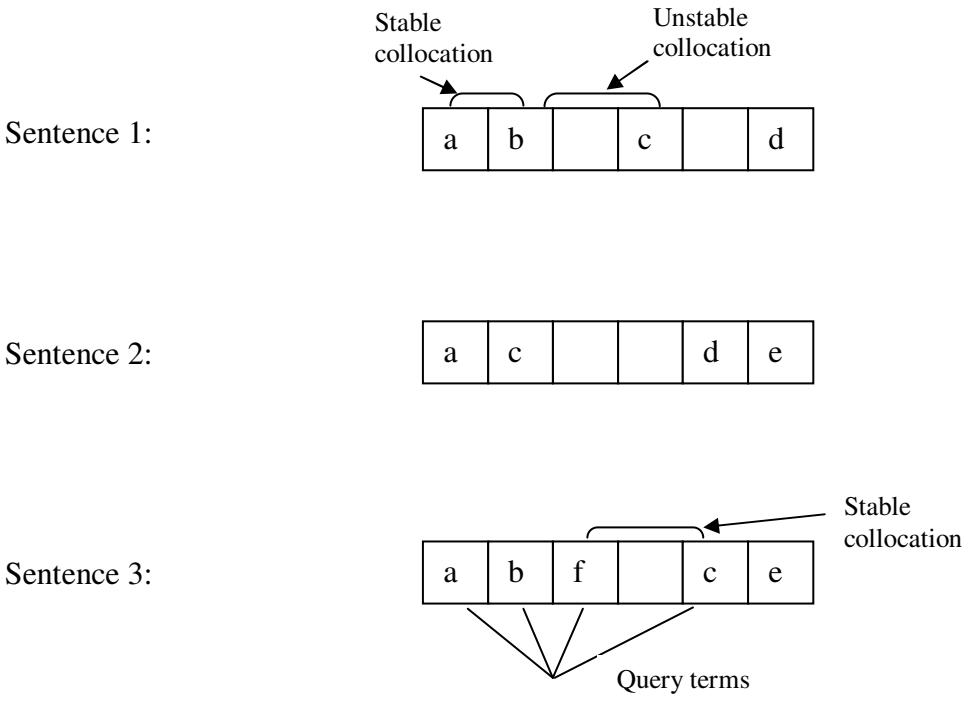

Figure 5.3 Stable and unstable collocation in a document

The stability of query terms collocation is determined according to the first inequality that is described in section 5.2 (Eq. 5.1). According to this equation, only query terms collocations that has a frequency variance greater than ten is a stable collocation and unstable otherwise.

Our calculation of a document matching score is the same as described in section 5.3.1, while our calculation of the contribution value is different and is computed as follows: for each query term instance, we find another query term instance that follow it in the same sentence and determine if they form a stable collocation or not. After determining stable query terms collocations in a sentence, the contribution value of the i^{th} instance of query term t in a sentence s is calculated according to the following formula (Eq. 5.9)

$$c(t_i) = \begin{cases} 1 + \dfrac{Bigrams(s)}{TotalBigram} \times Bonds(s) & if\ TotalBigram > 0 \\ 1 + \dfrac{Qterms(s)}{TotalQterms} \times Bonds(s) & if\ TotalBigram = 0 \end{cases} \quad (5.9)$$

$Bigrams(s)$ is the number of stable bigrams that are formed between consecutive query terms in sentence s.

$TotalBigram$ is the total number of stable bigrams that are formed between consecutive query terms in the query.

$Qterms(s)$ is the number of distinct query terms in a sentence.

$TotalQterms$ is the total number of query terms in a sentence.

$Bonds(s)$ is the number of bonds formed between sentence s and other sentences in a document. This relation is formed between sentences that have at least one lexical link formed between them. In this study, we only considered lexical link that are formed by simple lexical repetition.

6. Evaluation and Results

6.1 Evaluation in IR

A new proposed system or methodology for information retrieval must be evaluated. The evaluation process tries to measure the effectiveness of the system in providing the information that meets the user's need. Since testing information system with real users is time consuming and usually costly, the performance is usually measured by using a test collection, which consists of a set of documents, a set of topics, and a relevance judgment set for each topic. Figure 6.1 shows an example of a TREC topic. A relevance judgment set for each topic is constructed by trained annotators who take the role of users and then decide if the document is relevant to the query or not. One of the most popular and largest evaluation programs is TREC (Text REtrieval Conference), which has been established in 1992 and co-sponsored by the NIST (National Institute of Standards and Technology). To evaluate system performance, TREC provides both documents and queries for each participant. After running the queries by the proposed system, different evaluation measures are used to see how closely the retrieved documents match the documents in the relevance judgment set.

<num> Number 301
<title> International Organized Crime
<des> Description: Identify organizations that participate in international criminal activity, the activity, and if possible, collaborating organizations and the countries involved.
<narr> Narrative: A relevant document must as a minimum identify the organization and the type of illegal activity (e.g., Columbian cartel exporting cocaine). Vague reference to international drug trade without identification of the organization(s) involved not be relevant.

Figure 6.1 Example of a TREC topic

6.1.1 Evaluation Measures

The primary focus of IR research is to improve the retrieval performance. This performance is primarily related to how effective the system is in retrieving relevant documents as a response to the user query. There are several measures that were designed to evaluate the performance of an information retrieval system, e.g., Precision, Recall, Precision at top 10 retrieved documents (P10), R-precision, and Mean average Precision (MAP) (Baeza-Yates et al., 1999).

Precision: Is the proportion of retrieved documents that are relevant. It indicates how good an information system is in retrieving relevant documents. For instance, if all the documents that have been returned to the user are relevant, then precision is 100%. Having high precision is a good sign that the system retrieves the most relevant documents and omits non-relevant documents as much as possible. Precision also could be calculated by limiting the number of retrieved documents. For example, Precision at 10 (P@10) calculates precision in the top 10 retrieved documents instead of including all the retrieved documents. R-Precision on the other hand, has no predefined cut off but instead the cut off is R, which is the number of relevant documents in the relevant judgment set for particular topic. Average Precision is one of the most important measures, where precision is computed at every document found relevant, and the average of the precision values is calculated for each query.

Recall: Is another measure that calculates the proportion of relevant documents that are retrieved. It compares the number of the relevant documents that are retrieved by the proposed system with the number of relevant documents from the relevance judgment set. Recall is 100% when every relevant document has been retrieved. Even if we have 100% recall, it doesn't give

an idea if the system performance is good or bad. All the relevant documents might be at the bottom of the ranking, which is not satisfying since most users are more interested in those documents that are at the top of the ranking.

Many experiments in IR demonstrated that when the precision improved a noticeable decline in recall occurs and vice versa. Therefore, the best system is the one that would have the highest precision and recall at the same time. In this study, Precision at different document cut off, MAP, and R-precision will be used to evaluate the performance of our experiments.

6.2 Performance Evaluation and Results Discussion

Experiments were conducted based on the dataset from three TREC data collections. These collections are HARD2003, HARD2004, and HARD2005. Table 6.1 shows a statistical summary, including the number of topics and the number of documents that are associated with each data collection.

Collection	Number of documents	Number of topics
HARD2003 (no gov. docs)	321,405	50
HARD2004	635,650	50
HARD2005	1,036,805	50

Table 6.1 Number of topics and documents in each collection

Query terms were extracted from the "Title" field of TREC topics after removing all stopwords ("the", "of", "and", ….). Documents and queries have been stemmed using Porter Stemmer. Stopwords weren't removed from documents, since it affects terms positional information.

For each query in each collection, the top 2,000 documents were retrieved by using BM25 ranking function implemented in the Wumpus IR system (Büttcher et al. (2006)), where the values of the parameters b and k_1 are the one that showed the best performance in each data collection; and then these documents were re-ranked and the top 1,000 documents were retrieved by using one of the proposed methods (described in section 5.3). For each data collection we experimented with different values for the parameters (*b, k)* to determine their optimal values that will provide the optimal results for each experimental run. The optimal values of the parameters (*b, k*) of BM25 function that give the best results in our experimental runs for each data collection is given in the Appendix.

Regarding the first method, we experimented with different values for the parameter *C* (0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.75). In HARD2004, the optimal performance is obtained when the value of *C* is set to 0.3. In HARD2005, the optimal performance is obtained when the value of *C* is set to 0.75. While in HARD2003, the optimal performance is obtained when the value of *C* is set to 0.5.

MAP, Precision at 10 retrieved documents (P1@10), and R-precision evaluation measures have been used to evaluate the performance of our experimental runs. Table 6.2 shows the performance of our experimental runs in each collocation.

Method	Collection	P@10	MAP	R-prec
Method #1	HARD2004	0.3867	0.2422	0.2899
	HARD2005	0.4500	0.2149	0.2743
	HARD2003	0.5813	0.3435	0.3656
Method #2	HARD2004	0.3911	0.2522	0.2870
	HARD2005	0.4540	0.2078	0.2627
	HARD2003	0.5792	0.3340	0.3602

Table 6.2 Performance of the two experimental runs in the three collections

To further evaluate the performance of our methods, BM25-u ranking function and the two document ranking methods "Proximity" and "Bond" that are proposed by Vechtomova et al (2008) were used as baseline runs for comparison. BM25-u, as a ranking function, is the same as the BM25 implemented in Wumpus. The difference between the two functions is in stemming and implementation. While BM25 is implemented in Wumpus, BM25-u is implemented using perl scripts as in our experimental runs, so it is more appropriate to use it as a baseline run for comparison. Table 6.3 shows the best runs, which are achieved by BM25-u, "Proximity", and our first proposed method.

As can be seen from Table 6.3, In P@10 our run has a better performance than BM25-u in the three collections, while it only improves performance over "Proximity" in HARD2003. In MAP measure our run shows better performance than BM25-u in all three collections, while it only shows a better performance than "Proximity" in HARD2004. In R-prec our first run has a better performance than BM25-u in HARD2003, HARD2004 and HARD2005, and than "Proximity" in HARD2003, and HARD2004.

Collection	Run	P@10	MAP	R-prec
HARD2004	BM25-u	0.3689	0.2362	0.2861
	Proximity	0.3911	0.2394	0.2871
	Method #1	0.3867*	0.2422	0.2899
HARD2005	BM25-u	0.4420	0.2035	0.2639
	Proximity	0.4560	0.2150	0.2747
	Method #1	0.4500*	0.2149*	0.2743
HARD2003	BM25-u	0.5771	0.3383	0.3652
	Proximity	0.5708	0.3453	0.3647
	Method #1	0.5813	0.3435	0.3656

Table 6.3 Comparison between our first run and the two baseline runs in all three collections (* indicates that our first run is statistically significant compared to BM25-u at 0.05 significance level)

Table 6.4 shows the best runs, which are achieved by BM25-u, "Bond method", and our second proposed method. It shows that in HARD2004 our run provides a better performance than BM25 and "Bond" runs in all measures. In HARD2003 our run shows a better performance than BM25 only in P@10, and it has the same performance as "Bond" in P@10 and worse in MAP and R-prec. In HARD2005 our run has a better performance than BM25 in only two measures P@10 and MAP, and only better than "Bond" in one measure, which is MAP.

Collection	Run	P@10	MAP	R-prec
HARD2004	BM25-u	0.3689	0.2362	0.2861
	Bond	0.3711	0.2405	0.2839
	Method #2	0.3911	0.2522	0.2870
HARD2005	BM25-u	0.4420	0.2035	0.2639
	Bond	0.4580	0.2068	0.2678
	Method #2	0.4540	0.2078	0.2627
HARD2003	BM25-u	0.5771	0.3383	0.3652
	Bond	0.5792	0.3408	0.3636
	Method #2	0.5792	0.3340	0.3602

Table 6.4 Comparison between our second run and the two baseline runs in all three collections

To determine how significantly our runs outperform the baseline runs, t-test analysis at the significant level of 0.05 has been performed on all evaluation measures. As can be seen from Table 6.3 the performance of our first run is statistically significant over BM25-u in P@10 and MAP evaluation measures in HARD2005, while it is only significant in P@10 in HARD2004.

To further see how much improvement the two proposed methods have achieved over the baselines runs, we did a topic by topic comparison based on MAP. Figure 6.2 shows the comparison between our methods and BM25-u. As the figure show, out of 45 topics “method #1” improves performance in 19 queries, deteriorates performance in 10, and have the same performance in 16 queries; while “method #2” provides a better performance than BM25-u in 17 queries, worse in 18 queries and the same performance in 10 query.

As can be seen from figure 6.2, the amount of improvement or deterioration that is achieved by "method #1" over BM25-u is not significantly high in most of the topics; however a noticeable improvement has been obtained for topic 445.

Although the number of topics, whose performance are deteriorated by using "method #2", is higher than the number of topics that are improved, figure 6.2 shows that the total amount of improvement in MAP is higher than the total amount of deterioration; where "method #2" provides a noticeable performance improvement in topics 437 and topic 445, and a noticeable decline in performance in topic 447. By further looking at documents' ranking for these topics resulted from applying "method #2" and BM25-u, we notice that in topic 447 and 445 the ranking didn't change significantly at the top of the ranking. However, in topic 437 the ranking changed largely where two documents went up the ranking list from rank 66 and 47 to rank 1 and 2.

Figure 6.3 shows a topic-by-topic comparison between "method #1" and "Proximity" methods based on MAP. As seen from the figure, by applying "Method #1", the performance of 20 queries has improved, the performance of 14 queries has deteriorated and the performance of 11 queries was indifferent.

Figure 6.4 shows a topic-by-topic comparison between "method #2" and "Bond" methods based on MAP. From the figure, "method #2" improves the performance of 21 queries, deteriorate the performance of 14, and provide the same performance as "Bond" in 10 queries.

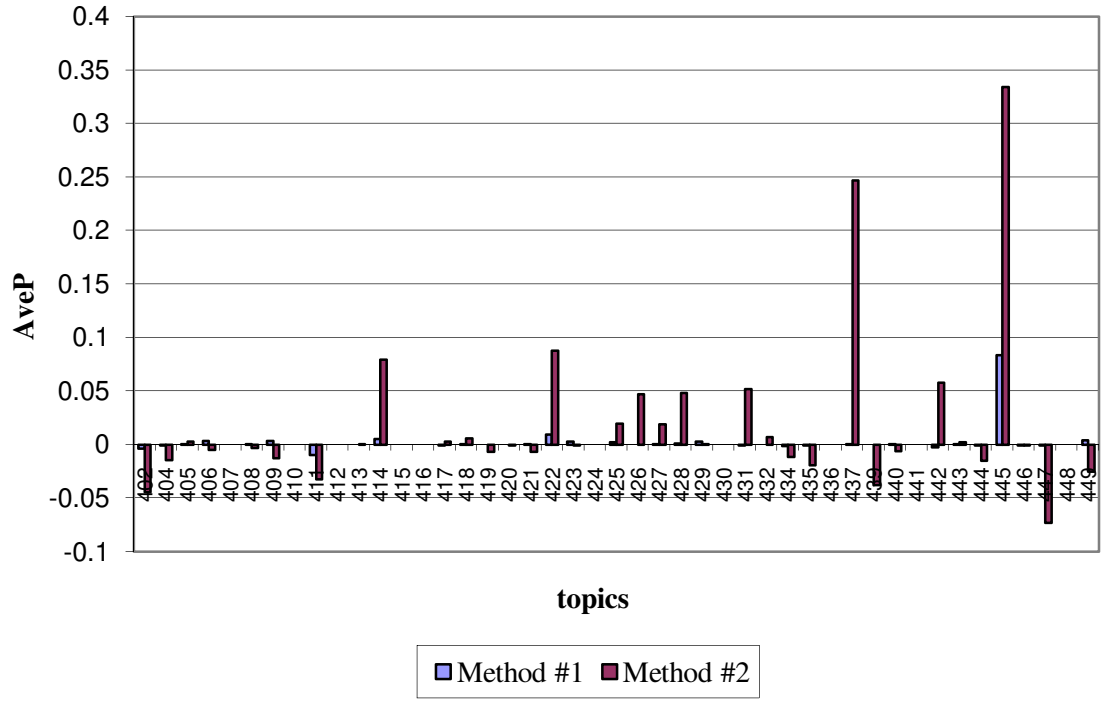

Figure 6.2 Topic-by-topic comparison between the two proposed methods and BM25-u (b=0.3, k1=1) based on MAP

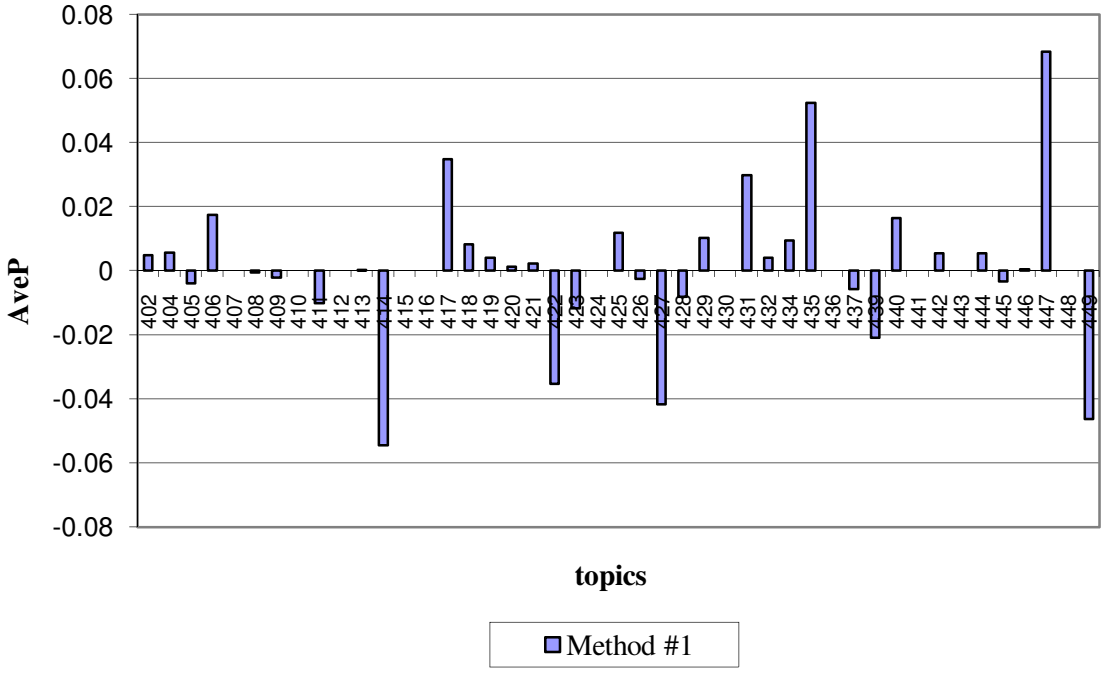

Figure 6.3 Topic-by-topic comparison between Method #1 and Proximity (b=0.3, k1=1) based on MAP

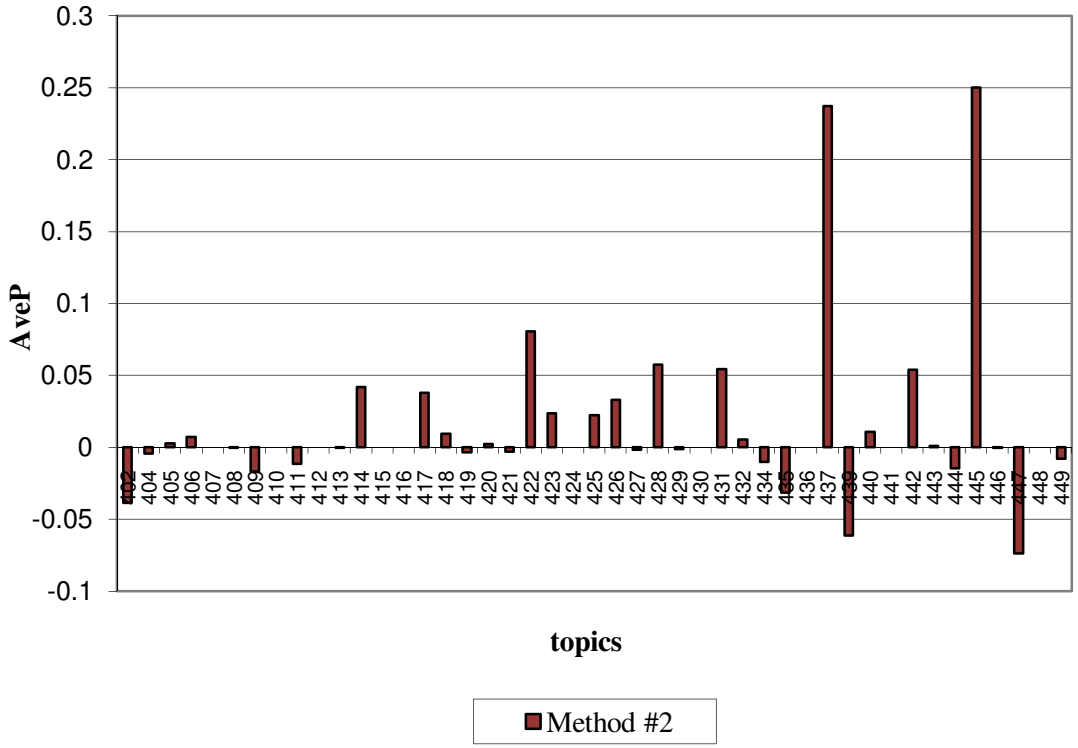

Figure 6.4 Topic-by-topic Comparison between Method #2 and Bond (b=0.3, k1=1) based on MAP

A more detailed look at some documents, which are demoted or promoted by using "method #2" over BM25-u, gives us an idea about the relationship between stable bigram collocation and document relevance. By looking at two documents that were promoted, we notice that almost half of the sentences in each document contain at least one query term and all query term bigrams, which are present in sentences that contain two or more query terms, are stable bigrams. For example, in topic 437 "role playing game", SLN20031223.0006 document was retrieved and promoted from the 66th rank in BM25 to the first rank in "method #2". Out of 119 sentences, 69 contain at least one query term, and all query term bigrams in these sentences are stable bigrams. The same is true for document NYT20030122.0369 that was retrieved for topic 402 "identity

theft". Out of the 32 sentences, 11 contain at least one query term and out of these sentences 10 contain the two query terms, where all the bigrams in these 10 sentences are stable bigrams.

For the documents that were demoted, we notice that most of these documents don't contain many query terms instances and very few sentences contain more than one query term. In topic 447 (vx nerve gas disposal), NYT20031013.0087 document contains 30 sentences, out of these 30 sentences 4 sentences contain one query term, 2 sentences contain two query terms, and no sentence contains 3 or 4 query terms. For the two sentences that contain 2 query terms, one of them contains stable bigrams and the other one doesn't. NYT20030808.0041 document in topic 447 (vx nerve gas disposal) is also demoted by "method #2". Out of the 43 sentences only 7 sentences contain at least one query term and out of these, 2 sentences contain 2 query terms, where one of these two sentences contain a stable bigram and the other doesn't. Out of the 7 sentences, no sentence contains 3 or 4 query terms.

The above analysis gives us an indication that "method #2" is more suitable for documents that contain many query terms instances, and where all the bigrams that are formed between query terms are stable bigrams. Therefore, further research could be done on the second part of equation (5.9), which is concerned with queries that contain no stable bigrams and with documents that contain no stable bigrams in their sentences.

7. Conclusion and Future Work

7.1 Conclusion

In this thesis, we propose two new document ranking techniques. These techniques integrates term proximity information (query terms collocations) into document ranking function. In these techniques, we distinguish between two types of collocation, collocation in the same grammatical structure and collocation in the same semantic structure. In the first method, query term's weight is affected by the occurrence of other query term in the same sentence. While in the second method, the terms weight is affected by the occurrence of other query term in the same sentence and in different sentence (Also known as transitive collocation).

Although our work is motivated by earlier studies of using term proximity information in document ranking (Clarke et al., 2000; Büttcher et al., 2006; Vechtomova et al., 2008), we depend in our methods on the positional frequency and stability of query terms co-occurrence rather than on the distance separating them, which was the focus of most of these studies.

In order to determine query terms collocation in the documents and query, we used a statistical collocation extraction technique, which identify collocation based on their co-occurrence frequencies (total frequency in the corpus within a predefined window size and terms positional frequency). This technique has been proven to be useful for identifying adjacent and non-adjacent query terms collocation, which makes it suitable for identifying query terms collocation that occur at a distance from each other.

We did our performance evaluation on three data collections (HARD2003, HARD2004, and HARD2005) by using three evaluation measures (P@10, MAP, and R-prec). The analysis of the results shows that our methods attained some improvement over the baseline runs either in all or some of these measures, although this improvement was not consistent in all three collections.

7.2 Future Work

In this study we show that using collocation in document ranking improve the retrieval results; however there is still a space for further improvement by taking the following consideration into account:

1- *The use of other association measures*: In this work, we didn't investigate the different methods of collocation extraction, such as mutual information, Z-score, and Log-Likelihood. Using such measures to identify collocation in document ranking may show a further improvement.

2- *Extending our proposed techniques to include n-gram collocations*: In identifying query term collocation, we only used the extraction technique for extracting bi-gram collocation; however this technique is also suitable for extracting n-gram collocation. Therefore, identifying n-gram query terms collocations and using them in document ranking may lead to further improvement.

3- *The use of syntactical information such as Part-Of-Speech (POS) to identify the stability of query terms collections*: In this study, collocations are identified based on a technique

that is introduced by Smadja (1993). Smadja technique integrates both statistical and syntactical information to determine collocation. However, in this study we only focused on using statistical information. Further improvement could be obtained by using syntactical information along with statistical information.

4- *The use of collocation extraction technique to determine the stability of co-occurrence relationship between query term and lexical link term*: When calculating lexical bond between sentences, we used simple lexical repetition; however early study by Vechtomova has indicated that the same lexical link term appear in relevant and non-relevant document. Therefore, the above collocation extraction technique could be used to determine the stability of co-occurrence relationship between query term and lexical link term; then based on this information, a word could be used as a link term or not.

Appendix: Experimental Runs Results

Run	HARD2004			HARD2005			HARD2003 (no gov.docs)		
Method #2	MAP	P@10	R-prec	MAP	P@10	R-prec	MAP	P@10	R-prec
k_1=1.5, b=0.5	**0.2522**	0.3778	0.2819	0.2057	0.448	0.26	0.3337	0.5604	0.3553
k_1=1.5, b=0.3	0.2513	0.3822	0.2762	**0.2078**	0.438	0.2617	0.328	0.5542	0.3554
k_1=1.5, b=0.6	0.2505	0.3756	0.2825	0.2045	0.442	0.26	**0.334**	0.5646	0.3588
k_1=2, b=0.3	0.2515	**0.3911**	0.2771	0.2063	0.444	0.258	0.3273	0.5604	0.3557
k_1=1.2, b=0.4	0.2512	0.38	0.2779	0.2075	**0.454**	0.2608	0.3311	0.5583	0.3524
k_1=2.5, b=0.75	0.2452	0.3667	0.2785	0.1984	0.418	0.2509	0.3267	**0.5792**	0.3568
k_1=0.5, b0.6	0.2463	0.3756	**0.287**	0.1968	0.432	0.2546	0.3191	0.5437	0.3396
k_1=1, b=0.5	0.2513	0.3711	0.2819	0.2057	0.448	**0.2627**	0.3286	0.5562	0.3481
k_1=2, b=0.6	0.2494	0.3778	0.2804	0.2031	0.43	0.257	0.3322	0.5646	**0.3602**
Method #1									
C=0.3, k_1=1.5, b=0.2	**0.2422**	0.3667	0.2782	0.2103	0.426	0.2654	0.3335	0.5646	0.3538
C=0.75, k_1=1.5, b=0.2	0.2404	0.3733	0.279	**0.2149**	0.432	0.2735	0.3342	0.5646	0.3506
C=0.5, k_1=1.2, b=0.5	0.2311	0.3489	0.259	0.2031	0.404	0.2601	**0.3435**	0.5625	0.3563
C=0.75, k_1=1, b=0.1	0.2366	**0.3867**	0.278	0.2058	0.432	0.266	0.3241	0.5583	0.3413
C=0.05, k_1=0.75, b=0.3	0.2324	0.3556	0.2742	0.1962	**0.45**	0.2588	0.3294	0.5604	0.3441
C=0.2, k_1=1, b=0.3	0.2337	0.3578	0.2801	0.2052	0.444	0.2644	0.3376	**0.5813**	0.3518
C=0.05, k_1=1.2, b=0.3	0.2372	0.3556	**0.2899**	0.2014	0.444	0.262	0.3356	0.5792	0.3504
C=0.75, k_1=2, b=0.2	0.2376	0.3756	0.2801	0.214	0.432	**0.2743**	0.332	0.5687	0.3522
C=0.05, k_1=1.5, b=0.6	0.2177	0.3378	0.2601	0.1922	0.388	0.2491	0.3403	0.5625	**0.3656**

References

1. Baeza-Yates, R., and Ribeiro-Neto, B. (1999). Modern Information Retrieval. New York, NY: ACM Press.
2. Benson, M. (1990) Collocations and General-Purpose Dictionaries, in International Journal of Lexicography 3: pp. 23-35.
3. Büttcher, S., Clarke, C.L.A., and Lushman, B. (2006). Term proximity scoring for ad-hoc retrieval on very large text collections. In: Proceedings of 29th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval.
4. Choueka, Y., Klein, T., and Neuwitz, E. (1983). Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus. ALLC Journal, vol. 4, pp. 34-38.
5. Choueka, Y. (1998). Looking for needles in a haystack or locating interesting collocations expressions in large textual databases. In Proceedings of the RIAO conference on User-Oriented Content-Based Text and Image Handling, Cambridge, MA.
6. Church, K., Gale, W., Hanks, P., and Hindle, D. (1991). Using statistics in lexical analysis. In Uri Zernik, editor, Lexical Acquisition: Exploring On-Line Resources to Build a Lexicon, pages 116-164. Erlbaum.
7. Church, K., and Hanks, P. (1989). Word association norms, mutual information, and lexicography. In Proceeding of the 27the meeting of the ACL, pages 76-83. Association for Computational Linguistics.
8. Clarke, C., Cormack, G., and Tudhope, E. (2000). Relevance ranking for one to three term queries. Information Processing and Management, 36:291–311.
9. Cleverdon, C. (1984). Optimizing convenient online access to bibliographic databases. Information Services and Use 4(1):37-47.

10. Fagan, J.L. (1989). The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. Journal of the American Society for Information Science, 40(2), pp. 115-132.
11. Fano, R. (1961). Transmission of information. MIT Press, Cambridge, Massachusetts.
12. Ferret, O. (2002). Using collocations for topic segmentation and link detection. In Proceedings of the 19th COLING.
13. Firth J.R. (1957): Papers in Linguistics 1934-1951. Oxford: Oxford University Press.
14. Halliday M.A.K., and Hasan R. (1976). Cohesion in English. Longman, London.
15. Hawking, D., and Thistlewaite, P. (1996). Relevance weighting using distance between term occurrences. Computer Science Technical Report TR-CS-96-08, Australian National University.
16. Hisamitsu, T., and Niwa, Y. (2002). A measure of term representativeness based on the number of co-occurring salient words. In Proceedings of the 19th COLING.
17. Hoey, M. (1991). Patterns of Lexis in Text. Oxford University Press.
18. Luhn, H. P. (1958). The automatic creation of literature abstracts, IBM Journal of Research and Development, 2:159-168.
19. Manning, C.D., and Schütze, H. (1999). Foundation of Statistical Natural Language Processing, MIT Press, Cambridge, Massachusetts.
20. McEnery, T., and Wilson, A. (1996). Corpus Linguistics, Edinburgh.
21. Metzler, D., and Croft B. (2005) A Markov Random Field Model for Term Dependencies. In Proceedings of the 28th ACM Conference on Research and Development in Information Retrieval SIGIR 2005, Salvador, Brazil, pp. 472-479

22. Mitra, M., Buckley, C., Singhal, A. and Cardie, C. (1997) an Analysis of Statistical and Syntactical Phrases. In Proceedings of RIAO-97, Montreal, Canada, pp. 200-214.

23. Morris, J. and Hirst, G. (1991) Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics, 17, pp. 21-48.

24. Rasolofo, Y., and Savoy, J. (2003). Term proximity scoring for keyword-based retrieval systems. In 25th European Conference on IR Research, ECIR 2003, number 2633 in LNCS, pages 207–218.

25. Robertson, S. E., and Spärck, J. K. (1976). Relevance weighting of search terms. Journal of the American Society for Information Science, 27, 129-146.

26. Salton, G. (1971). The SMART retrieval system: Experiments in automatic document processing. Prentice-Hall.

27. Smadja, F. (1993). Retrieving collocations from text: Xtract. Computational Linguistics, 19(1): 143-178.

28. Spärck, J, K., Walker, S., and Robertson, S. E. (1998). A probabilistic model of information retrieval: development and status, a Technical Report of the Computer Laboratory, University of Cambridge, U.K.

29. Van Rijisbergen, C. J. (1975). Information Retrieval. London, England: Butterworth & Co (Publishers) Ltd.

30. Vechtomova, O., Robertson, S., and Jones, S. (2003). Query expansion with long-span collocates. Information Retrieval, 6(2):251-273.

31. Vechtomova O., and Wang, Y. (2006). A Study of the Effect of Term Proximity on Query Expansion. Journal of Information Science, 32(4).

32. Vechtomova, O., and Karamuftuoglu M. (2008). Lexical Cohesion and Term Proximity in Document Ranking. Information Processing and Management, 44(4), pp. 1485-1502.