DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A NEW QUERY EXPANSION APPROACH USING CO-OCCURENCE BASED SEMANTIC RELATIONSHIP OF TERMS

by
Emre ŞATIR

August, 2015
İZMİR

A NEW QUERY EXPANSION APPROACH USING CO-OCCURENCE BASED SEMANTIC RELATIONSHIP OF TERMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of
Science in Computer Engineering

by
Emre ŞATIR

August, 2015
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"A NEW QUERY EXPANSION APPROACH USING CO-OCCURENCE BASED SEMANTIC RELATIONSHIP OF TERMS"** completed by **EMRE ŞATIR** under supervision of **ASSOC. PROF. DR. ADİL ALPKOÇAK** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Adil ALPKOÇAK

Supervisor

Yrd. Doç. Dr. Deniz KILINÇ

(Jury Member)

Yrd. Doç. Dr. Zerrin Işık

(Jury Member)

Prof.Dr. Ayşe OKUR
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First, I would like to thank to my supervisor, Assoc. Prof. Dr. Adil ALPKOÇAK, for his support, supervision and useful suggestions throughout this study.

I would like to also thank to Asst. Prof. Dr. Deniz KILINÇ for his continuous help and support during all phases of this research.

Finally, I would like to thank to Dr. Thomas K. Landauer from University of Colorado for providing TOEFL synonym dataset for our study.

I would like to dedicate this work to my wife Ferda ÖZKAYA ŞATIR for her support and patience during the development and writing of this thesis.

Emre ŞATIR

A NEW QUERY EXPANSION APPROACH USING CO-OCCURENCE BASED SEMANTIC RELATIONSHIP OF TERMS

ABSTRACT

In this thesis, we propose a new semantic-based query expansion approach, which uses co-occurrence information of words to acquire the semantics information of the terms. To do this, we first constructed word-context matrix using large English text corpus. Then we applied Singular Value Decomposition on the matrix to reduce the dimensionality. We validated the accuracy of semantic relationship on TOEFL synonym test.

Our query expansion approach uses this semantic relationship to improve the existing query expansion methods available in the literature, namely Bose-Einstein and Kullback-Leibler. We evaluated our approach on Milliyet collection, which is a Turkish IR test bed containing more than 400K documents and 72 queries. The experimentation shows that our approach clearly improves the all exsiting QE methods in terms of major IR performance measures such as MAP, r-precision and recall.

Keywords: Semantic, query expansion, information retrieval, word-context matrix, SVD.

TERİMLERİN BİRLİKTE GÖRÜNME TABANLI ANLAMSAL İLİŞKİLERİNİ KULLANAN YENİ BİR SORGU GENİŞLETME YAKLAŞIMI

ÖZ

Bu tezde, terimlerin anlamsal bilgilerini elde etmek için kelimelerin birlikte görünme bilgilerini kullanan yeni bir anlamsal tabanlı sorgu genişletme yaklaşımı öneriyoruz. Bunu yapmak için, ilk olarak büyük bir İngilizce külliyat kullanarak kelime-bağlam matrisi oluşturduk. Daha sonra boyutluluğu düşürmek için matris üzerinde Tekil Değer Ayrışımı uyguladık. Anlamsal ilişkileri TOEFL anlamdaş sözcükler testi üzerinde doğruladık.

Bizim sorgu genişletme yaklaşımımız, bu anlamsal ilişkileri, literatürde Bose-Einstein ve Kullback-Leibler isimleriyle bilinen sorgu genişletme yöntemlerini geliştirmek için kullanıyor. Yaklaşımımızı 400 binden fazla dokümandan ve 72 sorgudan oluşan bir Türkçe bilgi erişimi test yatağı olan Milliyet koleksiyonu üzerinde ölçtük. Deneyler gösteriyor ki, yaklaşımımız, MAP, r-precision ve recall gibi temel bilgi erişimi performans ölçütleri yönünden var olan bütün sorgu genişletme metodlarını açıkça iyileştiriyor.

Anahtar kelimeler: Anlamsal, sorgu genişletme, bilgi erişimi, kelime-bağlam matrisi,TDA.

CONTENTS

Pages

LIST OF FIGURES

Pages

LIST OF TABLES

Pages

CHAPTER ONE
INTRODUCTION

1.1 Overview

Information Retrieval (IR) has gained more importance with the Internet technology and the increase in the size of online text information. IR is a discipline that is related with indexing, retrieving, and structuring documents from any collections. The retrieval methods aim to find the best matching documents according to user query (user need) within a large document collection.

In general users may not know how to construct the best query according to their Information needs and the queries may be inadequate. Query Expansion (QE) is the process of reformulating the basic user query in order to get a better retrieving performance. Different techniques can be conducted to expand a query such as using synonyms of terms, using ontologies to add related terms, or checking spelling errors of terms and correcting them.

There are several ways of doing Query Expansion. For example expanded terms can be taken from outside of the corpus. But in our study we reformulate the queries with the words that are inside our corpus.

1.2 Purpose and Research Question

In this thesis, we propose a new Query Expansion approach, which uses word co-occurrence matrix to extract semantic relationship of terms. To prove that word co-occurence matrix is capable to extract synonymity of terms, we created word co-occurence on the British Nation Corpus (BNC) (British National Corpus, n.d.). and tested it with the Test of English as a Foreign Language (TOEFL) synonym test. (Landauer & Dumains, 1997). After we make sure that it works, we clearly formed our research question as follows; “if using word-corrcurence based semantic information in query expansion will improve retrieval performance”.

We compared our approach with existing query expansion methods available in the literature, namely Bose-Einstein and Kullback-Leibler (Perez-Aguera, & Araujo, 2008), which is already implemented in Terrier Information Retrieval Platform. To achieve this, we used Bilkent Milliyet Collection which is an IR test set containing more than 400K documents and 72 queries. (Can et al., 2008). The results we obtained from our experimentation shows that our approach clearly improves the all exsiting QE methods in terms of major IR performance measures such as MAP, R-precision and recall.

1.3 Thesis Organization

This thesis includes six chapters and the remaining of this thesis is organized as follows. In Chapter 2, previous works about our study is given. The works actually involve Information Retrieval, Query Expansion, Vector Space Models, Singular Value Decomposition, experiments on TOEFL synonym test, etc. In Chapter 3, some prior knowledge is given in detail. This information is necessary to understand our study. In Chapter 4, TOEFL synonym test experiment is described. First our data set (British National Corpus) is introduced and then we explain algorithm. This experiment can be seen a preliminary study for our query expansion studies. In Chapter 5, query expansion experiments are done. This chapter first gives information about our proposed system and then the results of the experiments are given. Finally in Chapter 6 the conclusion remarks and future works are given.

CHAPTER TWO
RELATED WORK

2.1 Literature Review

Our study consists of mainly two parts. In the first part, we evaluate the TOEFL synonym test. In the second part, we apply our technique to query expansion. In the literature there are several studies that are related to our work.

In their study (Can et al., 2008), researchers analyse the effects of several stemming options and query-document matching functions on retrieval performance by studying information retrieval (IR) on a large-scale Turkish text collection (they are the news articles and columns of five years, 2001 to 2005, from the Turkish newspaper Milliyet). They prove that three different stemmer approach provide similar retrieval effectiveness in Turkish IR. They also investigate the effects of a range of search conditions on the retrieval performance.

Turney and Pantel explain Vector Space Models (VSMs) in their paper (Turney & Pantel, 2010). They mention three types of matrix in a VSM, term-document, word-context and pair-pattern. They examine these three types and their applications. They say that term-document matrices are used for similarity of documents, word-context matrices for similarity of words and pair-pattern matrices for similarity of relations.

Researchers, in their study (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990) observed that they can shift the focus to measure word similarity, instead of document similarity, by looking at row vectors in the term–document matrix, instead of column vectors. They found an elegant way to improve similarity measurements with a mathematical operation on the term–document matrix based on linear algebra. The operation is truncated Singular Value Decomposition (SVD), also called thin SVD. Researchers briefly mentioned that truncated SVD can be applied to both document similarity and word similarity, but their focus was document similarity.

Schutze proposes to represent the semantics of words and contexts in a text as vectors (Schutze, 1992). He shows that dense vector representations are too difficult to use directly. So he suggests Singular Value Decomposition (SVD) as dimensionality reduction. He examines the structure of the vector representations and uses them for word sense disambiguation and thesaurus induction.

In a portion of his study (Rapp, 2003), to support his algorithm, Rapp counts word co-occurrences by using British National Corpus (BNC) (a 100-million-word corpus of written and spoken language). For counting word co-occurrences, a fixed window size is chosen. Based on that text window, he computes the co-occurrence matrix for the corpus. For the computation of semantic similarity, he uses the cosine coefficient and the city block metric. He also uses SVD for improving the results. As evaluation data, he uses Test of English as a Foreign Language (TOEFL), which comprises 80 test items. The accuracy of the evaluation study is 92.5% (74 of the 80 test items).

In their early seminal paper (Landauer & Dumains, 1997) Landauer and Dumains went the same way with the Rapp but they used a much smaller corpora (4.7 million words Grolier's Encyclopaedia). Also they didn't lemmatized the corpus and remove the function words. Finally their computations are based on a term-document matrix. They reported an accuracy of 64.4%.

Bullinaria and Levy (Bullinaria & Levy, 2007) take the road from the basic idea is simply that words with similar meanings will tend to occur in similar contexts, and so word co-occurrence statistics can provide a natural basis for semantic representations. For computing the word co-occurrence vectors, they use a text window of certain size. They present four evaluation tests and one of them is TOEFL. They utilize BNC corpus. They conclude that the best approach is Positive Pointwise Mutual Information (Positive PMI) with the cosine distance measure. They obtain a score of 85.0%, for the TOEFL task.

Turney presents a simple unsupervised learning algorithm for recognizing synonyms (Turney, 2001). The algorithm is called PMI-IR, which uses Pointwise

Mutual Information (PMI) and Information Retrieval (IR) to measure the similarity of word pairs. He applies the algorithm on TOEFL and English as a Second Language (ESL) and on both tests, the algorithm obtains a score of 74%. He compares the algorithm with the Latent Semantic Analysis (LSA), which achieves a score of 64% on the same 80 TOEFL questions.

Lund and Burgess (Lund & Burgess, 1996) have developed a related framework called HAL (hyperspace approximation to language). They use 160 million word corpus of USENET newsgroup text. They utilize a window representing a span of words with different sizes. Using the Euclidean distance between co-occurrence vectors they were able to predict the degree of priming of one word by another in a lexical decision task.

Matsuo and Ishizuka present a new keyword extraction algorithm that applies to a single document without using a corpus (Matsuo & Ishizuka, 2004). They say that, first frequent terms are extracted. Then co-occurrences of a term and frequent terms are counted. If a term appears frequently with a particular subset of terms, the term is likely to have important meaning. They claim that their algorithm shows comparable performance to tf-idf without using a corpus.

Stenmark, in his study (Stenmark, 2005), shows that intranet users like World Wide Web users submit very short queries resulting in unwanted results. He suggests Query Expansion (QE) techniques to improve their search results. He provides an automatic QE system based on Latent Semantic Indexing (LSI). He concludes that QE may be a useful technique on corporate intranets.

In their work (Perez-Aguera & Araujo, 2008), researchers test two approaches, co-occurrence approach and probabilistic approach, to extract the candidate query terms from the top ranked documents returned by the first-pass retrieval. They expand queries according to both approaches and compare the retrieval improvement. They also use a combined approach and get good results.

Researches, in their study (Amati, Carpineto & Romano, 2004), propose a selective application of QE to obtain a more robust retrieval system because although QE increases global Mean Average Precision (MAP), decreases the performance of worst (difficult) topics. Their objective is to avoid the application of QE on the set of worst topics. Their QE application predictor shows a performance similar to that of the unexpanded method on the worst topics, and better performance than full QE on the whole set of topics.

In our study, we use word-context matrix for similarity of words that is decribed in the study (Turney & Pantel, 2010). We apply SVD to our matrix like the studies (Deerwester, Dumais, Furnas, Landauer & Harshman, 1990), (Shutze, 1992), (Rapp, 2003), and (Landauer & Dumains, 1997). Also we evaluate the TOEFL synonym test in our study like the studies (Rapp, 2003), (Landauer & Dumains, 1997), (Bullinaria & Levy, 2007) and (Turney, 2001). They obtain different scores from the TOEFL test with different corporas and approaches.

CHAPTER THREE
PRELIMINARIES

3.1 Information Retrieval

As a result of rapid development of Internet and related technologies, the amount of data stored in computers is growing every day. This data has to be organized and retrieved whenever it is needed. Information Retrieval (IR) is accessing unstructured documents (e.g. text documents, images, and videos) that satisfy user need from within large collections (usually stored on computers) (Manning, Raghavan & Schutze, 2008).

In this thesis, *text retrieval* method is studied. This method covers retrieving from text document collections that contains words (also called terms). This is also called ad hoc retrieval, which the user provide a query and the system tries to find the documents, which satisfy best (Manning et al., 2008).

There are three main models for the text retrieval (Baeza-Yates & Ribeiro-Neto, 1999). These are the Boolean model, the probabilistic model and the vector space model. The Boolean model is a model that we can create queries with the Boolean operators AND, OR and NOT. The model views each document as just a set of words. In this model the frequency of a term in a document is not important. The main disadvantage of this model is that all returned documents are thought to have the same relevance. The retrieved documents are not ranked.

In the probabilistic and vector space models documents are ranked by relevance. The most basic measures (Manning et al., 2008) are the following:

- Precision: What fraction of the returned results are relevant to the information need?

$$precision = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} \tag{3.1}$$

- Recall: What fraction of the relevant documents in the collection were returned by the system?

$$recall = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} \tag{3.2}$$

Probabilistic model is beyond this thesis and interested readers can found information in (Baeza-Yates & Ribeiro-Neto, 1999) and (Fuhr, 1992). The vector space model, which is studied in this thesis will be explained in the next section.

3.2 Vector Space Model

In the vector space model, the documents and the query are represented as vectors. For a collection of *n* terms, the vectors are *n* dimensional. Each dimension corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. These values (i.e. term weights) can be computed in several different ways. One of the most popular is tf-idf (term frequency-inverse document frequency) weighting (Manning et al., 2008).

The tf-idf weighting scheme assigns to term *t* a weight in document *d* given by

$$tf - idf_{t,d} = tf_{t,d}\ x\ idf_t \tag{3.3}$$

In other words, $\text{tf-idf}_{t,d}$ assigns to term *t* a weight in document *d* that is

- highest when t occurs many times within a small number of documents
- lower when the term occurs fewer times in a document, or occurs in many documents
- lowest when the term occurs in virtually all documents.

For getting relevance scores, with a similarity measure, the query is compared with each document. Documents with the higher scores are considered as the most relevant. The most widely used similarity measures are Euclidean distance, dot product similarity and cosine similarity (Manning et al., 2008).

In their study (Turney & Pantel, 2010), Turney and Pantel examine Vector space models (VSMs) systems. They mention three types of matrix in VSMs: term-document matrices are used for similarity of documents, word-context matrices for similarity of words and pair-pattern matrices for similarity of relations.

3.3 Query Expansion

Query Expansion (QE) is the process of reformulating the basic user query in order to get a better retrieving performance. There are different techniques used in QE:

- Adding synonyms of words to the seed query
- Adding different forms of words (by stemming, lemmatization etc.) to the seed query
- Correcting spelling errors of the words in the seed query
- Changing the weights of the words in the seed query.

The motivation behind the QE is that the user might not provide the best query terms. Actually the methods about this problem split into two main classes: global methods and local methods (Manning et al., 2008). Global methods are techniques for reformulating query terms independent of the query and results returned from it. Global methods include:

- QE with an existing thesaurus
- QE via automatic thesaurus generation
- Techniques like spelling correction

Local methods reformulate a query according to the documents that initially retrieved from the seed query. The basic methods for this are:

- Relevance feedback
- Pseudo relevance feedback (Blind relevance feedback)

- Global (indirect) relevance feedback

Relevance feedback and pseudo relevance feedback are the most used and successful approaches. Because pseudo relevance feedback is used in this study, they will be presented in the next section.

3.3.1 Relevance feedback

The logic behind the relevance feedback is to involve the user in the retrieval process (Manning et al., 2008). The user gives feedback to the system about the relevance of the results. The user first gives the basic query and then the system returns the initial results. After that some documents are marked as relevant or nonrelevant by the user. And then the system reformulates the query according to the user feedback and then displays final results according to new query. The system can involve one or more iterations of this type.

3.3.2 Pseudo Relevance Feedback

Pseudo relevance feedback (Blind relevance feedback) uses a method like automatic local analysis (Manning et al., 2008). The user doesn't give feedback to the system, because the system automatically reformulates the query. First the system does the normal retrieval according to initial query and after finding documents the system assumes that the top *k* ranked documents are relevant, and finally to do relevance feedback as before under this assumption. Generally the procedure is:

- Take *k* documents from the initial retrieving as relevant results
- Select top *t* terms from these documents using for example tf-idf weights.
- Add these terms to query, do the retrieving process again and finally return new results.

3.3.3 Divergence From Randomness Model in Query Expansion

DFR (Divergence From Randomness) (Divergence From Randomness (DFR) Framework, n.d.) term weighting model infers the informativeness of a term by the

divergence between its distribution in the top-ranked documents and a random distribution (Perez-Aguera, & Araujo, 2008). Bo1 model is one of the most effective models that uses the Bose-Einstein statistics:

$$w(t) = tf_x \log_2 \left(\frac{1+P_n}{P_n}\right) + \log(1 + P_n) \quad (3.4)$$

where tf_x is the frequency of the query term in the x top-ranked documents and P_n is given by F/N , where F is the frequency of the query term in the collection and N is the number of documents in the collection.

Kullback-Liebler divergence computes the divergence between the probability distributions of terms in the whole collection and in the top ranked documents obtained for a first pass retrieval using the original user query (Perez-Aguera, & Araujo, 2008). The most likely terms to expand the query are those with a high probability in the top ranked set and low probability in the whole collection. For the term t this divergence is:

$$KLD_{(PR,PC)}(t) = P_R(t)\, log \frac{P_R(t)}{P_C(t)} \quad (3.5)$$

where $P_R(t)$ is the probability of the term t in the top ranked documents, and $P_C(t)$ is the probability of the term t in the whole collection.

3.4 Word-Context Matrix

Generally, when working with large collection of documents (i.e. large number of document vectors), it is logical to store these vectors into a matrix. The rows of the matrix correspond to terms and the columns correspond to documents. This kind of

matrix is called a term-document matrix. Figure 3.1 represents a simple example of a term-document matrix with 5 rows (5 terms) and 4 columns (4 documents).

	doc1	doc2	doc3	doc4
apple	1	0	1	1
banana	0	1	1	1
grape	0	0	1	0
orange	1	0	1	0
lemon	0	0	0	1

Figure 3.1 An example of term-document matrix.

Word similarity can be measured by looking at row vectors in the term-document matrix. But a document is not always the optimal length of text for measuring word similarity. So in general, we may have a word–context matrix, in which the context is given by words, phrases, sentences, paragraphs, chapters, documents, or more interesting possibilities, such as sequences of characters or patterns (Turney & Pantel, 2010).

The distributional hypothesis in linguistics claims that words, which occur in similar contexts tend to have similar meanings (Harris, 1954). A word may be represented by a vector in which the elements are derived from the occurrences of the word in various contexts, such as windows of words (Lund & Burgess, 1996). So, similar row vectors in the word-context matrix indicate similar word meanings.

3.4.1 Word Co-Occurrence Matrix and Sliding Window Concept

As a special use of word-context matrix, word co-occurrence matrix with a sliding window concept can be created. For a given collection of documents, a word-by-word co-occurrence frequency matrix can be generated via a fixed sized sliding window. All the words occurring within the window are considered as co-occurring with each other. By moving window across the documents, the co-occurrence matrix for all the words in a certain vocabulary is produced.

Two examples of word co-occurrence matrices from three example documents can be seen from the following figures (window size *n* means referring *n* words before and after the target word):

Document 1: A C B A D G A

Document 2: C D A B F

Document 3: F G C E

Vocabulary: A B C D E F G

	A	B	C	D	E	F	G
A	0	2	1	2	0	0	1
B	2	0	1	0	0	1	0
C	1	1	0	1	1	0	1
D	2	0	1	0	0	0	1
E	0	0	1	0	0	0	0
F	0	1	0	0	0	0	1
G	1	0	1	1	0	1	0

Figure 3.2 An example of word co-occurrence matrix from three example documents with window size 1.

	A	B	C	D	E	F	G
A	0	3	3	3	0	1	2
B	3	0	1	2	0	1	0
C	3	1	0	1	1	1	1
D	3	2	1	0	0	0	1
E	0	0	1	0	0	0	1
F	1	1	1	0	0	0	1
G	2	0	1	1	1	1	0

Figure 3.3 An example of word co-occurrence matrix from three example documents with window size 2.

3.5 Singular Value Decomposition

The basic idea behind the Singular Value Decomposition (SVD) is taking a high dimensional set of data points and reducing it to a lower dimensional space that indicates the substructure of the original data more clearly and orders it from most variation to the least. What makes SVD practical is that variation can be simply ignored below a particular threshold to reduce your data but be assured that the main relationships of interest have been preserved.

SVD is based on a theorem from linear algebra, which says that a rectangular matrix A can be broken down into the product of three matrices (Deerwester et al., 1990).

$$A = USV^T \tag{3.6}$$

such that U and V have orthonormal columns and S is diagonal. This is called the singular value decomposition of A. U and V are the matrices of left and right singular vectors and S is the diagonal matrix of singular values.

SVD also allows a simple strategy for optimal approximate fit using smaller matrices. If the singular values in *S* are ordered by size, the first *k* largest is kept and the remaining smaller ones are set to zero. This is called "Reduced Singular Value Decomposition", which is the mathematical technique underlying a type of document retrieval and word similarity method variously called "Latent Semantic Indexing" or "Latent Semantic Analysis". The logic behind this task is that it takes the original data, and breaks it down into linearly independent components. Because the very large part of those components is very small, they can be ignored, resulting in an approximation of the data that contains fewer dimensions than the original. SVD has the added advantage that in the process of dimensionality reduction, the representation of items that share substructure become more similar to each other, and dissimilar items become more dissimilar as well.

Consider the following matrix A:

$$A = \begin{matrix} 1 & 0 & 8 & 0 & 0 \\ 1 & 8 & 0 & 1 & 2 \\ 3 & 0 & 6 & 1 & 0 \\ 7 & 0 & 8 & 5 & 0 \\ 0 & 9 & 0 & 1 & 7 \end{matrix}$$

A is a 5x5 matrix. If it's wanted to choose the amount dimension reduction to 3, the following matrices are created:

$$U = \begin{matrix} -0.4522 & 0.1508 & -0.7704 \\ -0.1786 & -0.5494 & 0.1426 \\ -0.4281 & 0.1265 & -0.1752 \\ -0.7350 & 0.1827 & 0.5796 \\ -0.2003 & -0.7912 & -0.1400 \end{matrix}$$

$$S = \begin{matrix} 15.1349 & 0 & 0 \\ 0 & 13.7877 & 0 \\ 0 & 0 & 4.7957 \end{matrix}$$

$$V = \begin{matrix} -0.4665 & 0.0913 & 0.6054 \\ -0.2135 & -0.8353 & -0.0250 \\ -0.7973 & 0.2485 & -0.5375 \\ -0.2961 & -0.0218 & 0.5682 \\ -0.1162 & -0.4814 & -0.1450 \end{matrix}$$

CHAPTER FOUR
TOEFL SYNONYM TEST EXPERIMENT

In this chapter, we evaluate the TOEFL synonym test. It is introduced in Landauer and Dumais (Landauer & Dumais, 1997) as a way of evaluating algorithms for measuring degree of similarity between words. First we want to see if our technique works well with the synonym and then we apply it to query expansion. We used British Nation Corpus (BNC) is used (British National Corpus, n.d.) for our experiment. We chose BNC because it's a very large English corpus that reflects the widest possible variety of uses of the language.

Our study actually consists of two parts and following figures introduce the steps that we pass:

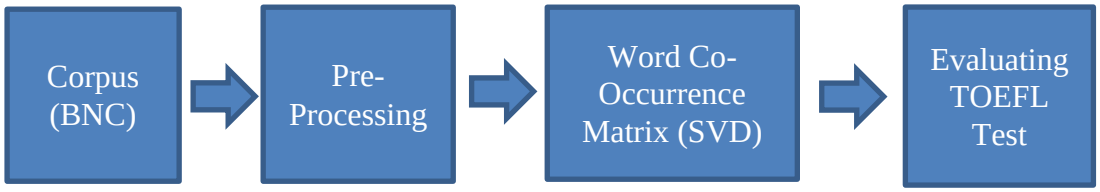

Figure 4.1 Flowchart of first part of our study.

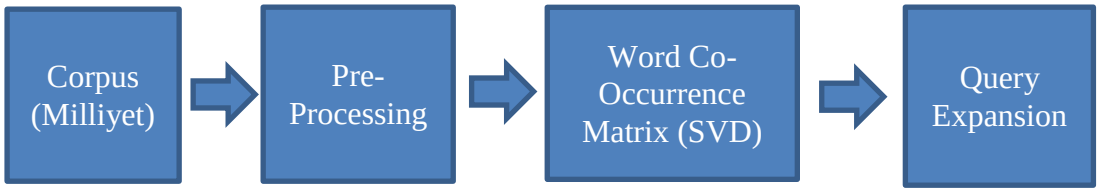

Figure 4.2 Flowchart of second part of our study.

4.1 Data Set

For the experiment, British Nation Corpus (BNC) is used (British National Corpus, n.d.). The British National Corpus is a collection of over 4000 samples of modern British English, both spoken and written, stored in electronic format. Over 100 million words in total, the corpus is currently being used by lexicographers to create dictionaries, by computer scientists to make machines understand and produce

natural language, by linguists to describe the English language, and by language teachers and students to teach and learn it and etc. It is not necessary to possess a copy of the corpus in order to make use of it: it can also be consulted via the Internet, using either the World Wide Web or the SARA software system, which was developed specially for this purpose (Aston & Burnard, 1998).

We obtained BNC XML edition from the University of Oxford Text Archive (The University of Oxford Text Archive, n.d.). The XML Edition is an enhanced and revised version of the original BNC World Edition, containing additional linguistic annotation, and many corrections.

4.1.1 Pre-Processing

The corpus that we obtained was in Text Encoding Initiative (TEI) format (TEI: Text Encoding Initiative, n.d.). There are 4049 documents in the corpus. To handle the corpus easily, first it was converted to free text. While selecting words, all uppercase letters were converted to their lowercase equivalents. The following two figures (Figure 4.1 and Figure 4.2) presents this pre-processing operation.

```
<w c5="AJ0" hw="immune" pos="ADJ">Immune </w><w c5="NN1" hw="deficiency" pos="SUBST">Deficiency </w><w c5="NN1" hw="syndrome" pos="SUBST">Syndrome</w><c c5="PUR">)</c></hi><w c5="VBZ" hw="be" pos="VERB">is </w><w c5="AT0" hw="a" pos="ART">a </w><w c5="NN1" hw="condition" pos="SUBST">condition </w><w c5="VVN" hw="cause" pos="VERB">caused </w><w c5="PRP" hw="by" pos="PREP">by </w><w c5="AT0" hw="a" pos="ART">a </w><w c5="NN1" hw="virus" pos="SUBST">virus </w><w c5="VVN" hw="call" pos="VERB">called </w><w c5="NP0" hw="hiv" pos="SUBST">HIV </w>
```

Figure 4.3 An example part from a document in BNC XML in TEI format.

```
immune deficiency syndrome is a condition caused by a virus called hiv
```

Figure 4.4 Converted text of the example TEI format document.

Secondly, we removed stopwords from the text. Stopwords are words, which are generally filtered out before processing of natural language text. Stopwords usually

refer to the most common words in a language. We used a 733 word English stopword list.

After stopwords removal, we made stemming operation on text. Stemming is the term used in information retrieval to describe the process for reducing inflected (or sometimes derived) words to their word stem, base or root form. A stemmer was written by Martin Porter and was published in the July 1980 (Porter, 1980). This stemmer was very widely used and became the de facto standard algorithm used for English stemming. We used Porter Stemmer Java program (The Porter Stemming Algorithm, n.d.) in our study. Example texts before and after stemming operation can be seen in Figure 4.3 and Figure 4.4.

future world health organisation projects 40 million infections year 2000 just beginning worldwide epidemic situation still unstable major impact come professor jonathan mann director who global aids programme acet international adviser useful contacts acet practical home care schools education training 081 840 7879 mildmay hospital hackney road london

Figure 4.5 An example text before stemming.

futur world health organis project 40 million infect year 2000 just begin worldwid epidem situat still unstabl major impact come professor jonathan mann director who global aid programm acet intern advis us contact acet practic home care school educ train 081 840 7879 mildmai hospit hacknei road london

Figure 4.6 An example text after stemming.

Finally all the numbers are removed from the text and the pre-processing step finished. The corpus is ready for processing.

4.2 TOEFL Synonym Test

Test of English as a Foreign Language (TOEFL) synonym test is an 80 multiple choice synonym questions and it has 4 choices per question. Figure 4.5 presents a sample question from the test:

question : levied
choices : (a) imposed
(b) believed
(c) requested
(d) correlated

solution : (a) imposed

Figure 4.7 A sample question from TOEFL synonym test.

For pre-processing, Porter Stemmer is applied on the TOEFL test data as described before. Figure 4.6 shows the sample question's state after stemming operation:

question : levi
choices : (a) impos
(b) believ
(c) request
(d) correl

solution : (a) impos

Figure 4.8 The sample question from TOEFL after stemming.

4.3 Creating the Word Co-Occurrence Matrix

As mentioned earlier a word-by-word co-occurrence frequency matrix can be generated via a fixed sized sliding window. We utilized the S-Space Package (The S-Space Package, n.d.) for creating this matrix. S-Space Package that is an open source framework for developing and evaluating word space algorithms. As a corpus, we used previously pre-processed BNC.

After running our application, it's seen that the total vocabulary is 251,898. It means that we have 251,898 unique terms. This is a very big number to process in the matrix. So we decided to restrict our vocabulary to all terms occurring at least 20 times in the corpus (i.e. we discarded terms that occurs in the corpus less than 20

times.). In this way our new vocabulary has 42,750 distinct terms. It means that we have a matrix of size 42,750x42,750. Different window sizes were tested and these experiments will be explained in the next section.

If dimensionality of the matrix is reduced before computing semantic similarities, results can be improved (Landauer & Dumais, 1997). This can be acghieved by using Singular Value Decomposition (Landauer & Dumais, 1997). Actually SVD is a dimension reduction technique that allows to significantly reduce the number of columns so smaller matrix has the advantage that all subsequent similarity computations run much faster. We studied different sizes of dimensionality reduction and these experiments can be seen in the next section.

4.4 Evaluating TOEFL Test

While evaluating the test, if one of the five words (question word or one of the four choices) is missing in the final version of the vocabulary (vocabulary of size 42,750) then we consider them as incomplete questions. There are six incomplete questions in our system. The question numbers are: 24, 32, 35, 49, 54 and 75. These questions of TOEFL test are ignored. So we evaluate the test as a 74 question test.

There are different types of similarity measures in the literature, herein; we use well-known cosine similarity measure. Actually the function we utilized is a distance function, which calculates one minus the cosine of the included angle between points. We go through as the following: If the distance result of vectors of question and answer words is less than (i.e. closer to zero) other choices' results, then it becomes a success. But if the distance result of vectors of question word and one of the choices is less than answer word's result, then it becomes a failure.

As with Rapp's study (Rapp, 2003), we decide to fasten the SVD dimension to 300 first. And then we change the window size between 1 to 10. The results can be seen in the Table 4.1:

Table 4.1 TOEFL evaluation results 1

Window size	Number of Success	Total Number of Questions	Accuracy (%)
1	54	74	72.97%
2	56	74	75.68%
3	56	74	75.68%
4	56	74	75.68%
5	59	74	79.73%
6	58	74	78.38%
7	57	74	77.03%
8	56	74	75.68%
9	56	74	75.68%
10	57	74	77.03%

As can be seen in the Table 4.1, window size 5 gives the best result. So for the second experiment we set the window size 5 and vary the SVD dimension between 100 to 1000. Table 4.2 presents the second experiment's results:

Table 4.2 TOEFL evaluation results 2

SVD Dimension	Number of Success	Total Number of Questions	Accuracy (%)
100	51	74	68.92%
200	58	74	78.38%
300	59	74	79.73%
400	60	74	81.08%
500	58	74	78.38%
600	57	74	77.03%
700	54	74	72.97%
800	57	74	77.03%
900	57	74	77.03%
1000	55	74	74.32%

Table 4.2 shows that we obtain the best result from the parameters that window size is 5 and dimension reduction parameter is 400. The accuracy of the best result is 81.08% (60 correct answers out of 74 questions).

4.5 Conclusion

Previous studies on TOEFL synonym test can be examined in the web site (TOEFL Synonym Questions, n.d.). Our best result 81.08% is in a good state in among others. But of course it can be improved. A different corpus might be used or a new pre-processing techniques can be developed. Also there are other dimensionality reduction techniques aside from SVD. Finally for evaluation there are several other similarity/distance metrics. These can be tested.

CHAPTER FIVE
PROPOSED APPROACH FOR QUERY EXPANSION

5.1 Data Set

In this study, for the query expansion experiment, Bilkent Milliyet Collection is used (Can et al., 2008). The collection contains 408,305 documents (news articles and columns of five years, 2001 to 2005) from Turkish newspaper Milliyet. Each document contains 234 words on the average without stopword elimination. The collection contains about 95.5 million words. The average word length is 6.90 characters.

There are also 72 ad-hoc queries that are evaluated by 33 assessors. Queries have fields that named topic, description and narrative. Topics are a few words that explains the query, descriptions are one or two sentences and narratives are more explanation about the query. In their study (Can et al., 2008) researchers name the queries as follows:

- Topic: short query
- Topic + Description: medium length query
- Topic + Description + Narrative: long query

5.2 Pre-Processing and Indexing

In information retrieval operations, pre-processing is an important step before indexing. Tokenization, stopword elimination and stemming are the most widely used pre-processing methods. We converted all uppercase letters to lowercase equivalents and then we converted Turkish special letters to their Latin alphabet counterparts (ç→c, ğ→g, ı→i, ö→o, ş→s , ü→u). For stemming, we used fixed prefix stemming (Can et al., 2008) and we selected the first 5 characters of a term as its stem. Researchers in their study showed that a simple word truncation approach provides similar performance in terms of effectiveness with other lemmatization

techniques (Can et al., 2008). We employed a semi-automatically generated stop-word list contains 147 words that taken from (Can et al., 2008). We indexed only headline and text fields of documents.

For indexing and retrieval operations, we used the Terrier IR Platform (The Terrier IR Platform, n.d.). Terrier is open source IR system written in Java and is developed at the School of Computer Science, University of Glasgow. It is a comprehensive, flexible and transparent platform for research and experimentation in text retrieval. We indexed headline and text fields of the documents by using Terrier. We performed tf-idf weighting model for our experiments.

5.3 Preparing the Word Co-Occurrence Matrix

As described in the Section 4.3, the word co-occurrence matrix is created via Milliyet Collection by utilizing S-Space Package. We took the headline and text fields of documents and before creating the matrix, we made some operations on the text as described below:

- Make lowercase letters (Astroloji Birliği → astroloji birliği).
- Omit the word parts after apostrophe (‘) (moda’da → moda).
- Remove midline (-) characters (e-posta → eposta).
- Remove all non-alphanumeric characters (şeklinde, özel → şeklinde özel).
- Remove all numbers (preserve digits in words) (ilkini 15 ocak mp3 → ilkini ocak mp3).
- Remove 1 length words (konuşmacı v seminer → konuşmacı seminer).
- Trim the extra whitespaces.
- Remove stopwords based on 147 elements stop word list (see the Section 5.2).
- Truncate the terms after the first 5 characters. If the word has less than 5 characters it remains the same (faaliyetlerini moda sürdüren astroloji birliği → faali moda surdu astro birli).
- Turkish special letters are converted to their Latin alphabet counterparts (ç→c, ğ→g, ı→i, ö→o, ş→s , ü→u).

Figure 5.1 and 5.2 present state of an example text before and after these operations, respectively.

Kasım ayından bu yana faaliyetlerini Moda'da sürdüren Astroloji Birliği Derneği kısaca "Astrobil" konuyla yakından ilgilenenler için yeni programlar tasarlıyor. Bunlardan ilki her ay tekrarlanacak olan "Gökyüzünde bu ay" toplantıları olacak. Girişin serbest olduğu ve 1,5 saat sürecek olan bu toplantılarda katılımcılar yaklaşan burcun bizlere neler getirebileceğini kendi görüşleri ve incelemeleri çerçevesinde aktaracaklar. Tamamen serbest konuşma şeklinde, özel bir formata gerek kalmaksızın astroloji öğrencileri, astrologlar konuşur ve tartışırken, konuya ilgi duyan arkadaşlar da bilgilerini artırmış, değişik yaklaşımları görmüş olacaklar.

Figure 5.1 Example text before text processing operations prior to matrix creation.

astro konus kasim ayind yana faali moda surdu astro birli derne kisac astro konuy yakin ilgil yeni progr tasar bunla ilki ay tekra gokyu ay topla olaca giris serbe saat surec topla katil yakla burcu bizle neler getir gorus incel cerce aktar tamam serbe konus sekli ozel forma gerek kalma astro ogren astro konus tarti konuy ilgi duyan arkad bilgi artir degis yakla gormu olaca

Figure 5.2 Example text after text processing operations prior to matrix creation.

After these text operations, we processed the converted collection with the application that we created using S-Space package. It's seen that total vocabulary (unique term size) is 139,916. SVD is a computationally exhausting operation so we decided not to use all terms in the vocabulary. We set the minimum frequency to 20 in our application, so we restricted our vocabulary to all terms occurring at least 20 times in the collection. So our new vocabulary size became 34,839 which led to a matrix of size 34,839x34,839. Also we determined the window size as five because, as can be seen from the previous chapter, we got the best result with this value. Finally we made SVD operation on this matrix with the reduction parameter of 400 (see section 4.4). So this gave us final matrix of size 34,839x400.

5.4 Proposed Approach

5.4.1 Query Expansion Mechanisms Implemented in Terrier

Terrier offers a query expansion functionality (Configuring Retrieval in Terrier, n.d.). Query expansion mechanism implemented in Terrier finds the most informative terms from the top-returned documents and adds these terms to the seed query. Terrier utilizes a particular DFR (Divergence From Randomness) (Divergence From Randomness (DFR) Framework, n.d.) term weighting model to weight terms in the top-returned documents. Currently, Terrier deploys the Bo1 (Bose-Einstein 1), Bo2 (Bose-Einstein 2) and KL (Kullback-Leibler) term weighting models. In default, these DFR weighting models follow a parameter-free approach.

In Terrier, there are two parameters that can be set for applying query expansion. The first one is the number of terms to expand a query with, which is 10 by default. The second one is the number of top-ranked documents from which these terms are extracted. The default value of this parameter is 3. The default values are used in all our experiments.

Following figures (Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6) present our first short query's weightings before and after QE implemented in Terrier:

kus gribi

Figure 5.3 First query (basic state).

kus^1.447727263 gribi^1.518950788 vietn^0.226569758 virus^0.095930707 hasta^0.092331876 vakas^0.075576062 h5n1^0.061793744 kisi^0.056443194 vakal^0.047738782 bulas^0.039179752

Figure 5.4 First query after the Bo1 weighting model.

kus^2.000000000 gribi^2.000000000 hasta^0.577028471 vietn^0.455508540 kisi^0.329933474 virus^0.243131151 soyle^0.198465158 yana^0.152678108 vakas^0.152678108 arali^0.105383424

Figure 5.5 First query after the Bo2 weighting model.

kus^1.570887490	gribi^1.626625605	vietn^0.240123781	hasta^0.157015615
virus^0.094987528	kisi^0.076277264	vakas^0.065976023	h5n1^0.050156352
vakal^0.039232325	yana^0.033843565		

Figure 5.6 First query after the KL weighting model.

5.4.2 Proposed Combined System

In this study, we propose a combined system, which includes QE mechanism implemented in Terrier and our word co-occurrence matrix based method. In this study we perform only short queries. The procedure that we follow is:

- Run Terrier with the one of the weighting models (Bo1, Bo2 or KL) and obtain the expanded terms.
- For every word in the basic query, run our word co-occurrence based system with the cosine threshold 0.9 (actually we use distance not similarity, i.e. one minus the cosine of the included angle between points). We choose 0.9 because we want to achieve most of the words that close to the seed word. We discard only words that are very far away from the seed word.
- Search for every word that Terrier found for the expanded terms in our word pool. If a word is included in both results, we added this word to the query (i.e. we filter out some words from QE mechanism implemented in Terrier.).
- Finally re-run Terrier with the original query and filter-outed expanded terms. We use the same weights taken from Terrier for the terms.

In the following figures (Figure 5.7 and Figure 5.8), an example procedure for an example query is presented. Our example query. Our example seed query is “kalitsal hastaliklar”. When we run Terrier with the Bo1 weighting method, it expands the query with some terms that can be seen in the Figure 5.8:

kalitsal hastaliklar

Figure 5.7 Example seed query.

belge kan tasiy ulusl almay hak sagli tedav

Figure 5.8 Expanded terms from example seed query.

When we look for the words “kalitsal” and “hastaliklar” in our matrix based system, the words “belge”, “ulusl” and “almay” are missing (i.e. with the threshold 0.9, we can’t find these terms in our system). So we filter-out these terms from the query. We expand the query only these words, so Figure 5.9 presents the final expanded terms:

kan tasiy hak sagli tedav

Figure 5.9 Final expanded terms.

We must mention that we use the same weights that Terrier supply us. The Figure 5.10 summarizes the procedure with the term weights (Bo1 weighting model is used.)

Seed query : kalitsal hastaliklar

Terrier’s new query with the weighting method Bo1 : kalit^1.172425226 hasta^1.177726624 belge^0.124236427 kan^0.064378896 tasiy^0.041000174 ulusl^0.031472054 almay^0.028465210 hak^0.028175312 sagli^0.024148779 tedav^0.022994547

Our combined system’s query : kalit^1.172425226 hasta^1.177726624 kan^0.064378896 tasiy^0.041000174 hak^0.028175312 sagli^0.024148779 tedav^0.022994547

Figure 5.10 Summary of the procedure with an example query.

5.5 Experimental Results

5.5.1 Baseline Results

For getting baseline results, we first performed Terrier with the following inputs and parameters:

- Corpus is 408,305 documents Milliyet Collection. We indexed only "headline" and "text" fields.
- We have 72 ad-hoc queries. We performed Terrier only with short queries (i.e. we used only "title" fields of the query file.). In this first experiment, the original queries are used with no query expansion terms.
- As a stopword list we used 147 words Turkish stopword list.
- Weighting model is tf-idf.
- For stemming, we used fixed prefix stemming and we selected the first 5 characters of a term as its stem.

We utilized trec_eval IR evaluation tool (Trec_eval, n.d.) for evaluation purposes. After using this tool, we got our baseline results for 72 queries. The following table (Table 5.1) shows the Mean Average Precision (MAP) result of our baseline.

Table 5.1 MAP result of the baseline

Query Length	**MAP**
short	0.3432

Table 5.2 presents the retrieved and relevant document numbers after evaluation process.

Table 5.2 Retrieved and relevant document numbers of baseline

Retrieved	70,099
Relevant	7,510
Rel_ret	5,807

The following table (Table 5.3) shows the interpolated recall-precision averages.

Table 5.3 Interpolated recall-precision averages of the baseline

at 0.00	0.7562
at 0.10	0.5716
at 0.20	0.5136
at 0.30	0.4466
at 0.40	0.3928
at 0.50	0.3501
at 0.60	0.3001
at 0.70	0.2452
at 0.80	0.1972
at 0.90	0.1300
at 1.00	0.0326

According the values in the table 5.3, precision-recall curve for the baseline is presented in the Figure 5.11:

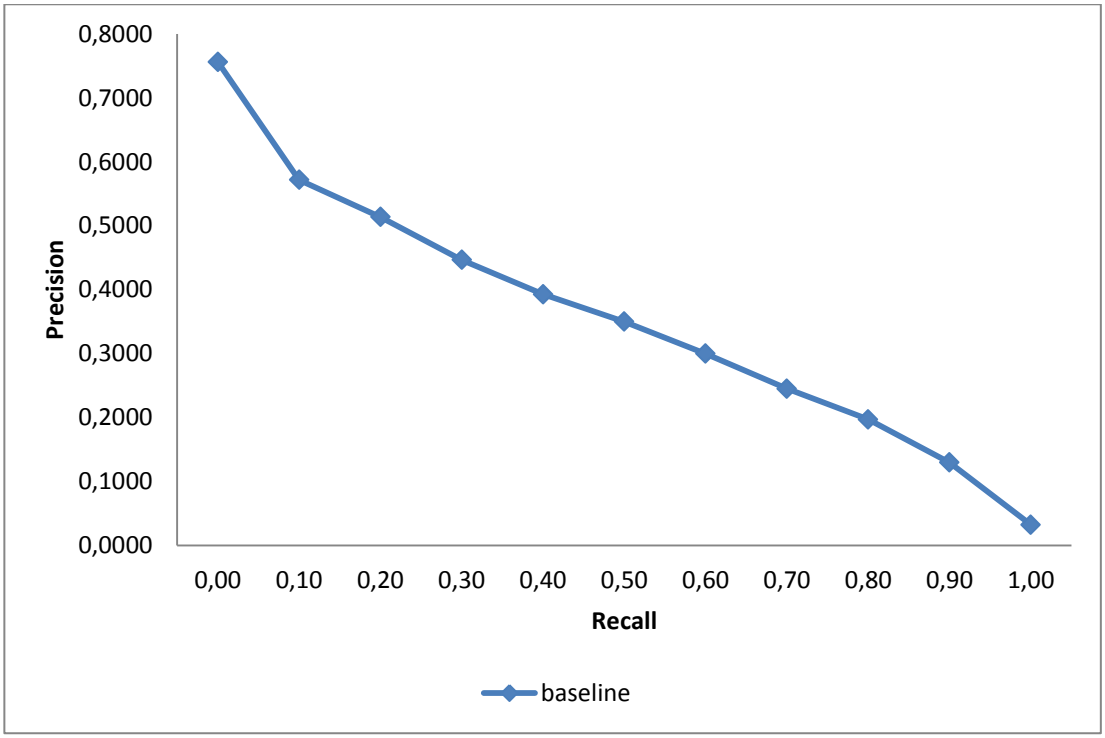

Figure 5.11 Precision-recall curve for the baseline

Table 5.4 presents the precision values at documents that retrieved.

Table 5.4 Precision values at documents of the baseline

at 5 docs	0.6028
at 10 docs	0.5667
at 15 docs	0.5398
at 20 docs	0.5243
at 30 docs	0.4940
at 100 docs	0.3754
at 200 docs	0.2636
at 500 docs	0.1408
at 1000 docs	0.0807

Finally R-Precision can be seen in the Table 5.5:

Table 5.5 R-precision of the baseline

Query Length	**R-Precision**
short	0.3613

5.5.2 QE Implemented in Terrier Results

For the second experiment, we run Terrier again with the same inputs and parameters but this time we used query expansion mechanism implemented in Terrier. As mentioned before, Terrier has some weighting models for the query expansion. We used Bo1, Bo2 and KL for our experiments.

Table 5.6 presents MAP results of the three QE weighting models.

Table 5.6 MAP results of the QE implemented in Terrier

Query Length	**MAP (Bo1)**	**MAP (Bo2)**	**MAP (KL)**
short	0.3652	0.3609	0.3663

As can be seen from the Table 5.6, the best MAP result is obtained from KL. Also all the MAP results are better than our baseline (which is 0.3432). So we can

comment that QE mechanism implemented in Terrier works well for our inputs and parameters.

Retrieved and relevant document numbers after evaluation process can be seen from the Table 5.7:

Table 5.7 Retrieved and relevant document numbers of QE implemented in Terrier

	Bo1	**Bo2**	**KL**
Retrieved	72,000	72,000	72,000
Relevant	7,510	7,510	7,510
Rel_ret	6,013	5,943	6,024

As can be seen from the Table 5.2, relevant retrieved documents number of baseline is 5,807. From the Table 5.7 we can obtain that this number is higher from the baseline for all three models.

Table 5.8 shows the interpolated recall-precision averages of three models:

Table 5.8 Interpolated recall-precision averages of three models

	Bo1	**Bo2**	**KL**
at 0.00	0.7646	0.7670	0.7640
at 0.10	0.6035	0.5990	0.6048
at 0.20	0.5390	0.5302	0.5416
at 0.30	0.4746	0.4689	0.4756
at 0.40	0.4197	0.4180	0.4219
at 0.50	0.3802	0.3749	0.3824
at 0.60	0.3263	0.3214	0.3279
at 0.70	0.2691	0.2665	0.2686
at 0.80	0.2127	0.2084	0.2141
at 0.90	0.1434	0.1380	0.1451
at 1.00	0.0336	0.0337	0.0341

Table 5.9 presents the precision values at documents that retrieved.

Table 5.9 Precision values at documents of three models

	Bo1	**Bo2**	**KL**
at 5 docs	0.5889	0.5861	0.5806
at 10 docs	0.5764	0.5653	0.5736
at 15 docs	0.5463	0.5509	0.5565
at 20 docs	0.5382	0.5271	0.5368
at 30 docs	0.5056	0.5069	0.5093
at 100 docs	0.3876	0.3739	0.3863
at 200 docs	0.2727	0.2653	0.2725
at 500 docs	0.1450	0.1444	0.1453
at 1000 docs	0.0835	0.0825	0.0837

R-precisions can be seen in the Table 5.10.

Table 5.10 R-precisions of three models

Query Length	**R-Precision (Bo1)**	**R-Precision (Bo2)**	**R-Precision (KL)**
short	0.3803	0.3742	0.3805

5.5.3 Proposed QE Results

As described in the Section 5.4.2, for the final experiment we performed QE mechanism implemented in Terrier and then we filter-out some words.

MAP results of the three combined QE models can be seen from the Table 5.11.

Table 5.11 MAP results of the combined methods

Query Length	**MAP (Combined-Bo1)**	**MAP (Combined-Bo2)**	**MAP (Combined-KL)**
short	0.3765	0.3713	0.3771

As can be seen from the previous tables and Table 5.11, MAP results of our combined system are higher from our baseline and QE mechanism implemented in Terrier.

Retrieved and relevant document numbers after evaluation process can be seen from the Table 5.12.

Table 5.12 Retrieved and relevant document numbers of combined systems

	Combined-Bo1	**Combined-Bo2**	**Combined-KL**
Retrieved	72,000	71,479	72,000
Relevant	7,510	7,510	7,510
Rel_ret	6,179	6,104	6,175

We can infer from the Table 5.7 and Table 5.12 that our combined system is better in getting relevant retrieved documents.

Table 5.13 shows the interpolated recall-precision averages of three combined models and improvements on the three models.

Table 5.13 Interpolated recall-precision averages of three combined models and improvements

	Bo1	**Com -Bo1**	**Imp. (%)**	**Bo2**	**Com -Bo2**	**Imp. (%)**	**KL**	**Com -KL**	**Imp. (%)**
at 0.00	0.7646	0.7891	3.20	0.7670	0.7954	3.70	0.7640	0.7869	3.00
at 0.10	0.6035	0.6174	2.30	0.5990	0.6203	3.56	0.6048	0.6206	2.61
t 0.20	0.5390	0.5616	4.19	0.5302	0.5469	3.15	0.5416	0.5601	3.42
at 0.30	0.4746	0.4886	2.95	0.4689	0.4806	2.50	0.4756	0.4900	3.03
at 0.40	0.4197	0.4377	4.29	0.4180	0.4260	1.91	0.4219	0.4382	3.86
at 0.50	0.3802	0.3963	4.23	0.3749	0.3871	3.25	0.3824	0.3933	2.85
at 0.60	0.3263	0.3419	4.78	0.3214	0.3339	3.89	0.3279	0.3389	3.35
at 0.70	0.2691	0.2766	2.79	0.2665	0.2720	2.06	0.2686	0.2767	3.02
at 0.80	0.2127	0.2162	1.65	0.2084	0.2217	6.38	0.2141	0.2182	1.91
at 0.90	0.1434	0.1437	0.21	0.1380	0.1445	4.71	0.1451	0.1479	1.93
at 1.00	0.0336	0.0372	10.71	0.0337	0.0379	12.46	0.0341	0.0379	11.14

The following three figures (Figure 5.12, Figure 5.13 and Figure 5.14) present precision-recall curves of the three combined models with baseline and QE implemented in Terrier:

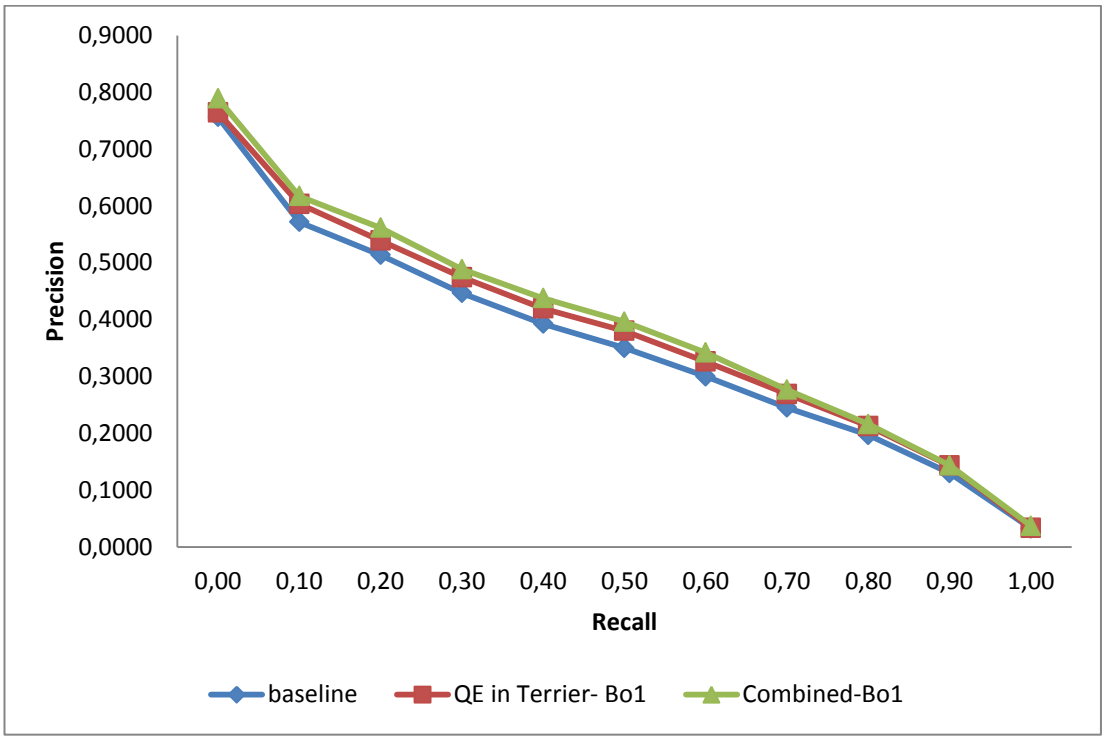

Figure 5.12 Precision-recall curve for the baseline, QE in Terrier-Bo1 and combined-Bo1.

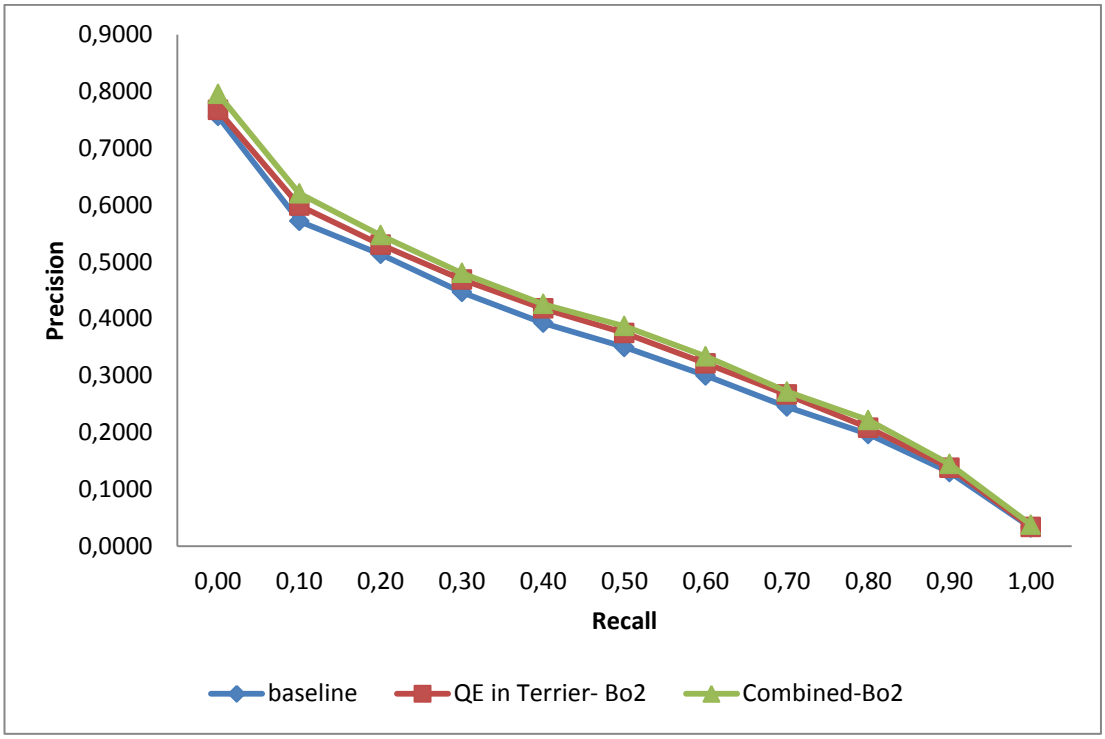

Figure 5.13 Precision-recall curve for the baseline, QE in Terrier-Bo2 and combined-Bo2.

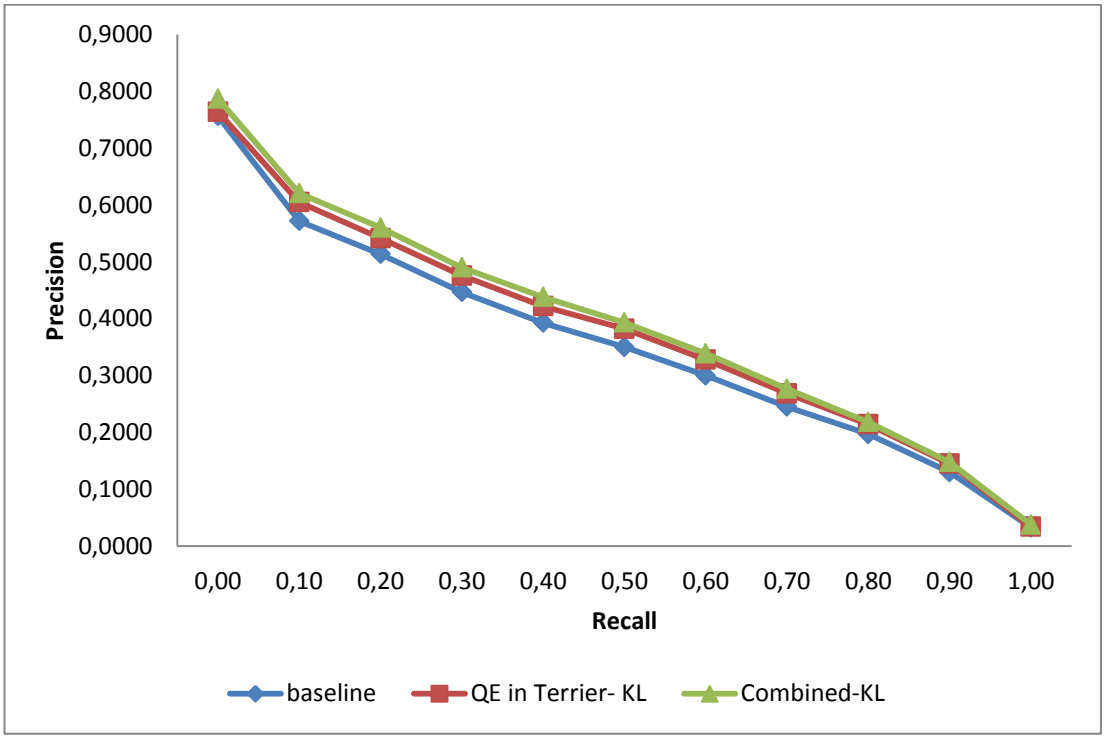

Figure 5.14 Precision-recall curve for the baseline, QE in Terrier-KL and combined-KL.

Table 5.14 presents the precision values at documents that retrieved for the combined models.

Table 5.14 Precision values at documents of three combined models

	Combined-Bo1	**Combined-Bo2**	**Combined-KL**
at 5 docs	0.6000	0.5944	0.6000
at 10 docs	0.5889	0.5889	0.5861
at 15 docs	0.5639	0.5657	0.5630
at 20 docs	0.5479	0.5514	0.5493
at 30 docs	0.5250	0.5222	0.5269
at 100 docs	0.3985	0.3846	0.3982
at 200 docs	0.2813	0.2723	0.2807
at 500 docs	0.1488	0.1486	0.1493
at 1000 docs	0.0858	0.0848	0.0858

R-precisions can be seen from the Table 5.15 for the three combined models.

Table 5.15 R-precisions of three combined models

Query Length	**R-Precision (Combined-Bo1)**	**R-Precision (Combined-Bo2)**	**R-Precision (Combined-KL)**
short	0.3926	0.3857	0.3934

CHAPTER SIX
CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this thesis, we proposed a new query expansion approach. We used co occurrence information of words to get their semantic relationship. We constructed a word co-occurrence matrix from the corpus. The approach does not require external knowledge; instead it uses the corpus itself. In other words, we develop ontology to expand the terms semantically using the corpus only. We investigate if co-occurrence information can help on the query expansion.

In the first part of our study, we achieved 81.08% success on the TOEFL synonym test with our word co-occurrence matrix based system using Singular Value Decomposition on the British National Corpus (BNC). In the second part, we applied this technique to perform Query Expansion on Turkish text (Milliyet corpus). We compared our approach with existing query expansion methods available in the literature, namely Bose-Einstein and Kullback-Leibler (Perez-Aguera, & Araujo, 2008), which is already implemented in Terrier Information Retrieval Platform. Experiments showed that our new approach produces better results for all metrics.

We can conclude that usage of co-occurrence information, collected from corpus via co occurence matrix, in Query Expansion improves the retrieval performance.

6.2 Future Work

Although we achieved to improve results nearly for all metrics in average, there are still queries that must be improved. Our technique can sometimes produce poor results on some queries. When we filter-out some words from the QE implemented in Terrier, some query results get worse. This is an issue that must be worked on.

For now we work only with single words. But plenty of queries have word phrases. Handling word phrases with the single words, can improve results fairly. To

do this, of course a mechanism must be developed to catch word phrases from the query words. Moreover, it can be further investigated the effects of different term weighting schemes in queries. i.e., the importance of words in the query must be determined.

REFERENCES

Amati, G., Carpineto, C., & Romano, G. (2004). Query difficulty, robustness and selective application of query expansion. *European Conference on IR Research, 2997*, 127-137.

Aston, G., & Burnard, L. (1998). *The BNC Handbook: Exploring the British National Corpus with SARA* (1st ed.). Edinburgh: Edinburgh University Press.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (1st ed.). New York: ACM Press.

British National Corpus, (n.d.). Retrieved December 16, 2014, from http://www.natcorp.ox.ac.uk/

Bullinaria, J.A., & Levy, J.P. (2007). Extracting semantic representations from word cooccurrence statistics: A computational study. *Behavior Research Methods, 39 (3)*, 510-526.

Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H.C., & Vursavas, O.M. (2008). Information retrieval on Turkish texts. *Journal of the American Society for Information Science and Technology, 59 (3)*, 407-421.

Configuring Retrieval in Terrier, (n.d.). Retrieved October 20, 2014, from http://terrier.org/docs/v3.6/configure_retrieval.html

Deerwester, S., Dumais, S.T., Furnas, G. W., Landauer, T.K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science (JASIS), 41 (6)*, 391–407.

Divergence From Randomness (DFR) Framework, (n.d.). Retrieved October 20, 2014, from http://terrier.org/docs/v3.6/dfr_description.html

Fuhr, N. (1992). Probabilistic models in information retrieval. *The Computer Journal, 35 (3)*, 243-255.

Harris, Z. (1954). Distributional structure. *Word, 10 (23)*, 146-162.

Landauer, T.K., & Dumais, S.T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review, 104 (2)*, 211-240.

Lund, K., & Burgess, C. (1996). Producing high dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers, 28 (2)*, 203-208.

Manning, C.D., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval* (1st ed.). Cambridge: Cambridge University Press.

Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools, 13 (1)*, 157-169.

Perez-Aguera, J.R., & Araujo, L. (2008). Comparing and combining methods for automatic query expansion. *Advances in Natural Language Processing and Applications Research in Computing Science, 33*. 177-188.

Porter, M.F. (1980). An algorithm for suffix stripping. *Program, 14 (3)*, 130-137.

Rapp, R. (2003). Word sense discovery based on sense descriptor dissimilarity. *Proceedings of the Ninth Machine Translation Summit*, 315-322.

Schutze, H. (1992). Dimensions of meaning. *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, 787-796.

Stenmark, D. (2005). Query expansion on a corporate intranet: Using LSI to increase precision in explorative search. *Proceedings of The 38th Hawaii International Conference on System Sciences.*

TEI: Text Encoding Initiative, (n.d.). Retrieved December 21, 2014, from http://www.tei-c.org/index.xml

The Porter Stemming Algorithm, (n.d.). Retrieved December 26, 2014, from http://tartarus.org/martin/PorterStemmer/

The S-Space Package, (n.d.). Retrieved April 12, 2015, from https://code.google.com/p/airhead-research/

The Terrier IR Platform, (n.d.). Retrieved September 15, 2012, from http://terrier.org/

The University of Oxford Text Archive, (n.d.). Retrieved December 16, 2014, from http://ota.ox.ac.uk/

TOEFL Synonym Questions, (n.d.). Retrieved January 15, 2015, from http://aclweb.org/aclwiki/index.php?title=TOEFL_Synonym_Questions_(State_of_the_art)

Trec_eval, (n.d.). Retrieved September 26, 2012, from http://trec.nist.gov/trec_eval/

Turney, P.D. (2001). Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the Twelfth European Conference on Machine Learning*, 491-502.

Turney, P.D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*, 141-188.