DOCUMENT RANKING BY GRAPH BASED LEXICAL COHESION AND TERM PROXIMITY COMPUTATION

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Hayrettin Gürkök

August, 2008

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. H. Murat Karamüftüoğlu(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. İbrahim Körpeoğlu

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. A. Yavuz Oruç

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

DOCUMENT RANKING BY GRAPH BASED LEXICAL COHESION AND TERM PROXIMITY COMPUTATION

Hayrettin Gürkök
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. H. Murat Karamüftüoğlu
August, 2008

During the course of reading, the meaning of each word is processed in the context of the meaning of the preceding words in text. Traditional IR systems usually adopt index terms to index and retrieve documents. Unfortunately, a lot of the semantics in a document or query is lost when the text is replaced with just a set of words (bag-of-words). This makes it mandatory to adapt linguistic theories and incorporate language processing techniques into IR tasks. The occurrences of index terms in a document are motivated. Frequently, in a document, the appearance of one word attracts the appearance of another. This can occur in forms of short-distance relationships (proximity) like common noun phrases as well as long-distance relationships (transitivity) defined as lexical cohesion in text. Much of the work done on determining context is based on estimating either long-distance or short-distance word relationships in a document. This work proposes a graph representation for documents and a new matching function based on this representation. By the use of graphs, it is possible to capture both short- and long-distance relationships in a single entity to calculate an overall context score. Experiments made on three TREC document collections showed significant performance improvements over the benchmark, Okapi BM25, retrieval model. Additionally, linguistic implications about the nature and trend of cohesion between query terms were achieved.

Keywords: Information retrieval, lexical cohesion, term proximity, collocation.

ÖZET

ÇİZGE TABANLI SÖZCÜKSEL BAĞDAŞIKLIK VE TERİM YAKINLIK HESABI İLE BELGE SIRALAMA

Hayrettin Gürkök
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. H. Murat Karamüftüoğlu
Ağustos, 2008

Okuma eylemi esnasında, her kelimenin anlamı, ondan önce gelen kelimelerin anlamları bağlamında işlenir. Geleneksel bilgi erişim sistemleri belgeleri tasnif etmek ve onlara erişmek için genellikle dizin terimleri kullanırlar. Fakat, metnin sıradan bir kelimeler kümesine dönüşmesi, belge ve sorgudaki anlamsal özellikleri de yok etmektedir. Bu durum, bilgi erişim işlemlerinde dilbilimsel teorileri uyarlamayı ve dil işleme tekniklerini uygulamayı mecbur kılmaktadır. Bir belgede dizin terimlerinin birlikte görülmesi tesadüfi değildir. Sıklıkla, bir belgede, bir kelimenin varlığı bir diğerinin varlığını çeker. Bu, tamlamalar gibi kısa mesafe (yakınlık) ya da sözcüksel bağdaşıklık olarak da adlandırılan uzun mesafe (geçişkenlik) ilişkisi şeklinde ortaya çıkabilir. Bağlam tespiti konusunda yapılan çoğu çalışma ya kısa ya da uzun mesafe sözcüksel ilişkileri tahmin etmeye dayanmaktadır. Bu çalışmada, belgeler için bir çizge gösterimi ve bu gösterime dayalı yeni bir sıralama sistemi önerilmektedir. Çizgeler yardımı ile, hem kısa hem de uzun mesafe sözcüksel ilişkileri tek bir yapıda tutup, belgeler için bir bağlam puanı hesaplamak mümkün olmaktadır. Üç TREC belge kolleksiyonunda yapılan deneyler, Okapi BM25 erişim modeline kıyasla önemli başarım artışı göstermiştir. Ayrıca, belgelerde bulunan sorgu terimleri arasındaki bağdaşıklığın doğası ve eğilimi hakkında dilbilimsel sonuçlar elde edilmiştir.

Anahtar sözcükler: Bilgi erişimi, sözcüksel bağdaşıklık, terim yakınlığı, eşdizimlilik.

Acknowledgement

This thesis serves as a tribute to my advisor, Asst. Prof. Dr. H. Murat Karamüftüoğlu, for the time, patience, and effort he has spent on me. My M.S. education could not have begun, nor would be completed without his initiative. I am indebted for the vision, knowledge, and mentality I acquired from him and I feel privileged and proud to have benefited from his mentoring and guidance.

I am grateful to Dr.-Ing. Markus Schaal for the discussions we had which helped a lot in shaping of this study and for his support anytime I needed. I am thankful to the members of my jury, Asst. Prof. Dr. İbrahim Körpeoğlu and Prof. Dr. A. Yavuz Oruç, for the honor of reviewing and approving the quality of this work. I would also like to thank my fellow Cihan Öztürk for the time he spent in proofreading the whole text.

Finally, many thanks to my beloved parents for their everlasting support which motivated me through challenges and made it possible for me to complete this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Information Retrieval (IR)

The science of IR is concerned with the representation, storage, organization of, and access to information items [4]. By the increasing amount of digital information becoming available every day, fast access to these resources becomes even more difficult. This also adversely affects the ability to reach the 'correct' information. IR research tries to mitigate these problems in order to provide in the best way the information which might be relevant or useful to the user.

It is useful to clarify some IR terminology before starting discussion. The records that IR addresses are called *documents*. Documents are retrieved from an organized and relatively static repository, most commonly called a *collection* (also called *archive* or *corpus*). IR is not restricted to static collections though. For instance, the collection may be a stream of messages flowing over the Internet [11]. User's representation of information need is called *query*, which is generally textual, and the words in the query are called *keywords*.

In a simplistic IR system there are three components: input, processor and output (Figure 1.1 from [39]). Most computer-based retrieval systems store only a representation of the document (or query) which means that the text of a

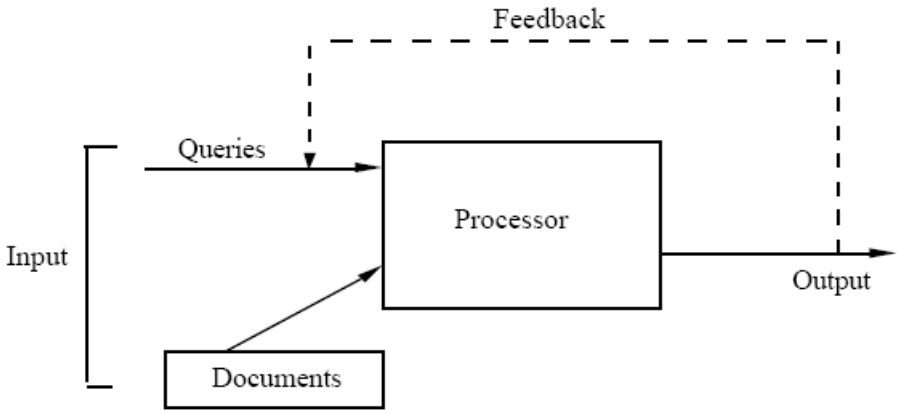


Figure 1.1: A typical IR system

document is lost once it has been processed for the purpose of generating its representation. For example, a document representative could be a list of extracted words considered to be significant. The words in the original document, which are processed and transformed to the document representative are now called *terms*. It is possible for the user to change his request during one search session in the light of a sample retrieval to improve the subsequent retrieval run. Such a procedure is referred to as *feedback*. The processor is concerned with structuring the information in an appropriate way and executing the search strategy in response to a query. The output is usually a set of citations or document numbers referring to documents deemed relevant by the IR system [39].

1.2 IR Performance Evaluation

One of the primary distinctions made in the evaluation of IR systems is between effectiveness and efficiency. *Effectiveness* measures the ability of the search engine to find the right information, and *efficiency* measures how quickly this is done [7]. Due to the purpose of this work, retrieval effectiveness is considered as the performance indicator.

1.2.1 Measuring IR Effectiveness

The major goal of an IR system is to retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible. Relevance is an inherently subjective concept [35]. People often disagree about whether a document is related to a given query or not. The disagreement is more prominent if "degree of relevance" is considered, rather than "absolute relevance". Moreover, a person can be in disagreement even with himself due to different needs, preferences, knowledge, expertise, language, and etc. Relevance may also depend on the collection a document is retrieved from or the order it is presented [11].

Three items are required to measure IR effectiveness [20]:

1. A document collection
2. A test suite of information needs, expressible as queries
3. A set of relevance judgments, standardly a binary assessment of either relevant or non-relevant for each querydocument pair.

1.2.2 Standard Test Collections

To address the three requirements mentioned in §1.2.1, *standard test collections* consisting of documents, queries, and relevance judgments were assembled by researchers. Using test collections provide various advantages. Firstly, given the large size of collections, it is very difficult to ask real users to assess the relevance of answer sets consisting of hundreds of documents to each different query. Secondly, considering the number of different combinations an IR system's parameters might produce, it is impractical to conduct relevance judgment sessions with real users for tuning purposes.

There are numerous standard test collections. A well-known and still updated collection series is maintained by *TREC* (*Text REtrieval Conference*). TREC is

a workshop series designed to build the infrastructure necessary for the large-scale evaluation of text retrieval technology. The series is sponsored by the U.S. National Institute of Standards and Technology (NIST) and the U.S. Department of Defense [45]. At the time of this writing, there have been sixteen TREC workshops. A variety of retrieval tasks (*tracks*) on different collections were introduced in TREC. In total, TREC test collections comprise six CDs containing 1.89 million documents (mainly, but not exclusively, newswire articles) (Figure B.8) and relevance judgments for 450 information needs, which are called *topics* and specified in detailed text passages (Figure B.9) [20].

Relevance judgments require considerable manual effort for high-recall search tasks. While for small collections most of the documents in the collection could be evaluated for relevance, in today's large collections this would clearly be impossible. Instead, a technique called *pooling* is used. In this technique, the top k results (for TREC, k varied between 50 and 200) from the rankings obtained by different search engines (or retrieval algorithms) are merged into a pool, duplicates are removed, and the documents are presented in some random order to the people doing the relevance judgments [7]. Pooling is good for producing large number of relevance judgments for each query. Its limitation is that, if a document is found relevant by a new algorithm but it was not part of the pool, it will be treated as non-relevant and the effectiveness of that algorithm could be significantly underestimated. Ingwersen defines this situation as the *Dark Matter problem* of IR and describes it as follows: "the searcher, the IR system, and the IR researcher, 'does not know what he does not retrieve' - and will never know it" [18]. However, studies with the TREC data have shown that the relevance judgments are complete enough to produce accurate comparisons for new search techniques [7].

It is wrong to report results on a test collection that were obtained by tuning parameters to maximize performance on the same collection. Such a tuning overstates the expected performance of the system, as the parameters will be set to maximize performance on one particular set of queries rather than for a random sample of queries. In such cases, the correct procedure is to have one or more *development test collections* and to tune the parameters on the development test

collection. Then the tester would run the system with those parameters on the *test collection* and reports the results on that collection as an unbiased estimate of performance [20].

1.2.3 IR Effectiveness Metrics

There are two major retrieval effectiveness metrics, precision and recall. *Precision* is the fraction of retrieved documents that are relevant and *recall* is the fraction of relevant documents that are retrieved. Recall measures the ability of the system to retrieve useful documents while precision measures the ability to reject useless materials [35]. Formally:

$$Precision = \frac{\#(relevant\ items\ retrieved)}{\#(retrieved\ items)} \tag{1.1}$$

$$Recall = \frac{\#(relevant\ items\ retrieved)}{\#(relevant\ items)} \tag{1.2}$$

Another metric standard among the TREC community is *mean average precision* (MAP), which provides a single-figure measure of quality across recall levels. For a given query, average precision is the average of the precision value obtained for the set of top k documents existing after each relevant document is retrieved, and this value is then averaged over number of queries. If the set of relevant documents for a query $q_j \epsilon \mathrm{Q}$ is $\{d_1, ..., d_{m_j}\}$ and R_{jk} is the set of ranked retrieval results from the top result until document d_k is reached, then [20]:

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \tag{1.3}$$

For many applications what matters is how many good results there are on the first (few) page(s). This leads to measuring precision at fixed low levels of retrieved results, such as ten or thirty documents. This is referred to as *precision at k* (e.g. precision at 10). Another alternative metric is *R-precision*, which is the same as "precision at k" with k = (number of relevant documents).

1.2.4 Significance Tests

Once the retrieval effectiveness figures are obtained, in order to decide whether this data shows that there is a meaningful difference between two retrieval algorithms, *significance tests* are needed. Croft et al. proposes the following procedure for comparing two retrieval algorithms using a particular set of queries and a significance test [7]:

1. Compute the effectiveness measure for every query for both rankings.
2. Compute a *test statistic* based on a comparison of the effectiveness measures for each query. The test statistic depends on the significance test, and is simply a quantity calculated from the sample data that is used to decide whether or not the null hypothesis should be rejected.
3. The test statistic is used to compute a *P-value,* which is the probability that a test statistic value at least at that extreme could be observed if the null hypothesis were true. Small P-values suggest that the null hypothesis may be false.
4. The null hypothesis (no difference) is rejected in favor of the alternate hypothesis (i.e. B is more effective than A) if the P-value is $\leq \alpha$, the *significance level*. Values for α are small, typically 0.05 and 0.1, to minimize Type I errors.

So, if the probability of getting a specific test statistic value is very small assuming the null hypothesis is true, we reject that hypothesis and conclude that ranking algorithm B is more effective than the baseline algorithm A [7].

1.3 Classic IR Models

In classical IR models, each document is described by a set of representative keywords called index terms. An *index term* is simply a (document) word whose

semantics helps in remembering the main themes of the document. Index terms are used in indexing and summarizing document contents. Index terms are mainly nouns which have meaning by themselves so that their semantics is easier to identify and grasp compared with adjectives, adverbs, and connectives which function mainly as complements [4].

Within a set of index terms for a document, not all terms are equally useful for describing the document contents. For instance in a collection of hundred thousand documents, a term appearing in each document is useless as an index term because it does not tell anything about which documents the user might be interested in. On the other hand, a word appearing in very few documents is quite useful narrowing the space of documents which might be of interest to the user. Distinct index terms have varying relevance when used to describe document contents. This effect is captured through the assignment of numerical *weights* to each index term of a document [4]. Weights can also be assigned to the terms in a query. The weight of a query term is usually a measure of how much importance the term will be assigned in computation of the similarity of documents to the given query. Weights are usually normalized to be fractions between zero and one [11].

1.3.1 Boolean Model

Boolean model is a simple retrieval model based on set theory and Boolean algebra. It considers that index terms are either present or absent in a document. This implies that term weights are assumed to be all binary (i.e. 0 or 1). The query is formulated as a Boolean combination of keywords using operators *and*, *or*, and *not*. For example, a query 'k_1 *and* k_2' is satisfied if and only if a document contains both keywords k_1 and k_2. More complex queries can be built out of these basic operators to be evaluated using Boolean algebra [4].

It is possible to make refinements on a classic Boolean query. First, the query can be applied to a specific syntactic portion of the document, like title or abstract, instead of the whole document. Second, a position to apply the

query can be specified, like the beginning of the title of a document [11]. Another possibility is to incorporate an *adjacency* operator, say *adj*, to the operator set. So the result of a query 'k_1 *adj* k_2' will ensure that k_1 and k_2 are contained in adjacent word positions. This is helpful in searching for phrases like 'information retrieval' [35]. The adjacency operator can be extended to a *proximity* operator which may be used to specify that two terms must be within n words (or sentences) of each other (e.g. n=0 may mean that the words must be adjacent). A proximity operator can be applied to Boolean conditions as well as to simple terms. For instance it might specify that a sentence satisfying one Boolean condition must be adjacent to a sentence satisfying some other Boolean condition. A proximity operator may specify order as well as proximity. It may define not only how close two words must be but in what order they must occur [11].

Boolean model is an *exact matching* model, which means that a document either satisfies a query or not. Since there is no grading scale, ranking is not possible. This leads to answer sets consisting of either too few or too many documents which prevent good retrieval performance.

1.3.2 Vector Space Model

Vector space is a statistical model which recognizes the disadvantages associated with the Boolean model. It allows *partial matching* by assigning non-binary weights to index terms in queries and documents. These term weights are then used to compute the degree of *similarity* between documents and query. This allows documents to be ranked more precisely [4].

Given a system with t index terms, vector space model considers a query q and each document in the collection d_j as t-dimensional vectors $\overrightarrow{d_j}$ and $\overrightarrow{q}$. It evaluates the degree of similarity between the query and the document ($sim(d_j,q)$) according to the correlation between their corresponding vectors by a *matching function*. There are many examples of matching functions in the literature. One

of them is taking the cosine of the angle between query and document vectors [4]:

$$sim(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \times |\vec{q}|} \tag{1.4}$$

Various methods for assigning weights to index terms were suggested. Some alternatives can be found in Salton and Buckley's paper [33]. One early idea is to use *inverted document frequency* (*idf*) defined by Spärk Jones [36]. This weighting scheme sorts the terms in reverse order according to the number of documents in a collection in which the term occurs. So, terms occurring in many documents receive low weights. If N is the number of documents in a collection and n_k is the number of documents in which term k occurs, then the inverse document frequency of term k, idf_k, is defined as [34]:

$$idf_k = logN/n_k \tag{1.5}$$

The most commonly used weighting scheme is *tf-idf* (*term frequency-inverted document frequency*) weighting. This is calculated as a combination of two values:

1. A value based on collection occurrence of the index term, idf (Eqn. 1.5).
2. A value based on document occurrence of the index term. Frequency of occurrence, also known as *term frequency* (*tf*), of a term can be used to compute this value.

Finally the tf-idf weight, $tfidf_{ik}$, of term k in document i can be defined as [34]:

$$tfidf_{ik} = idf_k \cdot tf_{ik} \tag{1.6}$$

The disadvantage of vector space model is that it considers the index terms mutually independent. This comes along with the advantage of making it a simple and fast model. Due to the locality of many term dependencies, their indiscriminate application to all the documents in the collection might in fact badly affect the retrieval performance [4].

1.3.3 Probabilistic Model

According to the probabilistic model, given a user query there exists a set containing exactly the relevant documents and no other (ideal set). Provided there is an exact description of this ideal set, the retrieval will be ideal too. The probabilistic model starts with an initial guess of probabilistic description of the ideal set to retrieve the initial set of relevant document. By interacting with the user, the description of the ideal set is improved [4].

The original and still most influential probabilistic retrieval model is the *binary independence model* (*BIM*) [28]. Here, *binary* means that if a term is present in a document (or query) it is represented by 1 in the document (or query) vector and by 0 otherwise. *Independence* means that terms are modeled as occurring in the document independently. The model recognizes no association between terms [20]. This model has the advantage of sorting the documents according to their probability of being relevant. However, it suffers from considering index terms as independent, not weighting terms by frequency of occurring inside a document (i.e. all weights are binary), and requiring an initial guess for describing the ideal set [4].

Based on the BIM, the *F4* weighting formula was developed. For a document i, provided the relevance information is available, the F4 formula is [30]:

$$w_i = log \frac{(r + 0.5)(N - R - n + r + 0.5)}{(R - r + 0.5)(n - r + 0.5)} \tag{1.7}$$

where

N = collection size
n = number of postings of the term
R = total known relevant documents
r = number of these posted to the term

The matching function is a simple sum-of-weights.

The F4 was later elaborated by its originators. In proceedings of TREC-3

Robertson et al. stated that the original F4 model (Eqn. 1.7) was with "no account taken of document length or term frequency within document or query" [32] and developed two models, BM11 and BM15, in which "the simple inverse collection frequency term-weighting scheme (F4) was elaborated to embody within-document frequency and document length components, as well as within-query frequency" [32]. These two models were described in TREC-2 proceedings [31]. In TREC-3 they introduced a new model, *BM25*, which is a combination of BM11 and BM15 models [32]. According to the BM25 model the weight of a term i in a document D is calculated as [37]:

$$W(TF_i) = \frac{TF_i\,(k_1+1)}{K+TF_i}\,w_i \tag{1.8}$$

where

$K = k_1 * ((1-b) + b\frac{DL}{AVDL})$
$k_1,\ b$ = tuning constants
DL = length of D (i.e. number of terms in D)
$AVDL$ = average document length in the given collection
w_i = Eqn. 1.7
TF_i = frequency (number of occurrences) of i in D

The matching score for the document is the sum of the weights of the matching (i.e. present) terms. Robertson et al. identify three characteristics of the BM25 weighting formula (Eqn. 1.8) [37]:

1. It is zero for $TF_i = 0$,
2. It increases monotonically with TF_i,
3. When $TF_i = 1$ the weight is just the usual presence weight w_i. Additional occurrences of t_i increase its contribution to the score, but there is an absolute limit on how much they can add (has an asymptotic limit).

The constant k_1 determines how much the weight reacts to increasing TF. If $k_1 = 0$, the weight reduces to the term-presence weight only; if k_1 is large, the

weight is nearly linear in *TF*. It was found to have values in the range 1.2 - 2 to be effective. [37].

The formula given by K is for document length normalization. If the tuning constant b is set to 1, the simple normalization factor is used. Smaller values reduce the normalization effect. Experiments with the TREC collection suggest a value of around $b = 0.75$ is good [37].

1.4 Problem Statement

In contrast with data retrieval systems which just determine the documents containing the keywords in a user's query, IR systems aim to retrieve *information* about a subject in order to satisfy the user's need. Van Rijsbergen states that the 'perfect' retrieval might be achieved by a human being reading an entire collection of documents to satisfy a query in hand retaining the relevant documents and discarding all the others, but this is obviously impractical [39]. It is not only the physical or timing constraints but also the much superior interpretation capability of a human versus an automatic IR system which causes this impossibility. 'Reading' involves attempting to extract information, both syntactic and semantic, from the text and using it to decide whether each document is relevant to a particular request or not.

During the course of reading, the meaning of each word is processed in the *context* of the meaning of the preceding words in text. Van Rijsbergen emphasizes that "If a document contains information about X then it is likely to be relevant to X ... The process of locating relevant documents (however), is inherently uncertain, it is also highly context dependent. The uncertainty enters in a number of ways, first through the aboutness, (where) it is only possible to determine that a document is about something to a degree, hence our probabilistic models, secondly, whether a document is relevant to an expressed need is also a matter of degree. Finally, a document is about X with the probability α, it may or may not contain the information X" [40].

As described in §1.3 traditional IR systems usually adopt index terms to index and retrieve documents. Unfortunately this is an oversimplification of the problem because a lot of the semantics in a document or query is lost when the text is replaced with just a set of words (*bag-of-words*). However the occurrences of index terms in a document are motivated. Frequently, in a document, the appearance of one word attracts the appearance of another. This can occur in forms of short-distance relationships (proximity) like common noun phrases as well as long-distance relationships (transitivity) defined as lexical cohesion in text, to be explained in the next chapter.

None of the classic IR models described in §1.3 considers the interaction between the words in a document but rather they are regarded as independent entities. Not exploiting the lexical-semantic relationships between the words of a document limits the retrieval effectiveness due to the reasons explained in §1.4. This makes it mandatory to adapt linguistic theories and incorporate language processing techniques into IR tasks.

Much of the work done on determining context is based on estimating either long-distance (§2.2) or short-distance (§2.3) word relationships in a document. These are covered in detail in the next chapter. This work proposes a graph representation for documents and a new matching function, CGS, based on this representation. By the use of graphs, it is possible to capture 'both' short- (by direct paths between query terms) and long-distance (exploiting transitive paths between query terms) relationships in a single body to calculate an overall context score which will increase retrieval effectiveness.

By the advantage of using graphs and calculating the cohesion score in stages (of path, pair, and document scores), it is possible to observe the relationship between lexical collocation patterns and cohesion in text.

In addition, the graph representation can be used to visualize the document contents so as to display the document words, index terms, and the connections between words which may facilitate easy content analysis and relevance judging. The scores calculated according to the new CGS matching function can be used as an input to existing information visualization tools. An example can be seen

in Figure B.7 where the graph representations of relevant and non-relevant documents for the same topic are visualized using a graph visualization tool, Chisio [19].

The thesis is organized as follows. In Chapter 2, the previous work on linguistic cohesion and its applications to IR are presented. The details of graph-based document ranking methods developed in this work are given in Chapter 3. Chapter 4 describes the experimental setup used in evaluation of the methods presented. The results of the evaluation experiments are given and discussed in Chapter 5. Chapter 6 summarizes the experimental results and points to future research directions.

Chapter 2

Related Work

2.1 Linguistic Cohesion

The methods proposed in this thesis are based on linguistic theories and hypotheses developed on cohesion therefore it is useful to introduce these here. Text is made of meanings expressed in words and structures. It is essentially a semantic unit itself; it is wrong to consider it as a bigger version of sentence.

Every text is a context for itself and is characterized by coherence; "it hangs together" [14]. Hoey defines *coherence* as "a quality assigned to text by a reader or listener, and is a measure of the extent to which the reader or listener finds that the text holds together and makes sense as a unity. It is therefore not identifiable with any combination of linguistic features and will never be absolute. The same text may be found coherent by one reader and incoherent by another, though an overwhelming consensus can be achieved for most naturally-occurring texts." [16]. Hasan also claims that "textual coherence is a relative, not an absolute property" [15].

An important feature that facilitates coherence is *cohesion*, a set of linguistic resources that every language has for linking one part of a text to another. These linguistic resources (or *cohesive ties*) are divided into five classes which are

conjunction, reference, substitution and ellipsis, and lexical cohesion [14].

Conjunction is the author's use of adjunct-like elements to mark semantic relationships between the sentences. Items like 'however', 'alternatively', and 'on the other hand' may all serve to mark a perceived semantic relation. *Reference* does not 'mark' semantic relations; it 'is' a semantic relation and occurs whenever an item indicates that the identity of what is being talked about can be retrieved from the immediate context. Reference items include pronouns and determiners [16]. In the following sentence the words typed in bold are a determiner and a pronoun respectively referring to a car: 'There appeared a car. **The** car was so fast that **it** disappeared in the nick of time'. *Substitution and ellipsis* are grammatical relations; the former occurs when a class of items stands in for an earlier lexical item in the text, the latter when what stands in for the earlier item is nothing at all [16]. The sentence 'I play the cello. My husband does, too.' demonstrates an example of substitution where the word 'does' replaces 'play'. And the sentence 'Yes, you can borrow my pen but what happened to yours?' is elliptical where 'yours' is used in place of 'your pen' [14].

Initially, Halliday and Hasan defined *lexical cohesion* loosely as various kinds of semantic relationships between lexical items. A categorization of these relationships was then made by Hasan. The sub-categories she recognizes are given in Table 2.1, taken from [15]:

Category	Sub-category	Example
A. General	a. repetition	leave, leaving, left
	b. synonymy	leave, depart
	c. antonymy	leave, arrive
	d. hyponymy	travel, leave
	e. meronymy	hand, finger
B. Instantial	a. equivalence	the *sailor* was their *daddy*
	b. naming	the *dog* was called *Toto*
	c. semblance	the *deck* was like a *pool*

Table 2.1: Categories of lexical cohesion

Having made the distinction between coherence and cohesion, one might expect that it would be computationally easier to identify cohesion, because the

identification of ellipsis, reference, substitution, conjunction, and lexical cohesion is a straightforward task for people. Halliday and Hasan's analysis on seven texts of a variety of kinds reveals that lexical cohesion accounts for over forty per cent of cohesive ties. Table A.1 shows the distribution of each class of tie per text [13]. This high frequency of occurrence makes lexical cohesion a strong candidate for determining the cohesion in text.

Morris and Hirst [24] showed that lexical cohesion is computationally feasible to identify. A single instance of a lexical cohesive relationship between two words is usually referred to as a *lexical link*. Morris and Hirst state that lexical cohesion does not only occur between pairs of words but over a succession of a number of nearby related words spanning a topical unit of the text. They call these sequences of related words as *lexical chains*. They claimed that since lexical cohesion is a result of a unit of text being about a single topic, and text structure analysis involves finding the units of text that are about the same topic, one should have something to say about the other. They proved this by computing lexical chains on general-interest magazine articles and showing that these correspond closely to the intentional structure produced from the structural analysis method of Grosz and Sidner [12].

Hoey [16] introduced the concept of *lexical bonds* defined as the connection that exists between a pair of sentences by virtue of there being an above-average number of links relating them. He argues that the minimum number of links required is three (and it is never less than three) but sometimes for texts in which there are a great number repetitions, the threshold may be four links or more. He claimed that bonded pairs of sentences are semantically related and, often, intelligible together.

2.2 Lexical Cohesion in IR

There are a number of works on usage of lexical cohesion in information retrieval most of which are based on computing lexical chains or lexical bonds. Stairmand [38] developed an IR system which identifies lexical clusters and lexical chains of semantically related terms using WordNet [22] synonym sets (synsets) and then quantifies textual contexts by considering the distribution of these terms throughout the document. During retrieval, for each query concept they establish its context of occurrence, and then determine how dominant this textual context is within the document based on a vector-space model. They compared their system, COATER, against IBM's STAIRS retrieval system and demonstrated performance improvement. However they also noted that recall performance was limited by the coverage of the WordNet database, thus making the system incapable of being compared with standard test collections.

Ellman and Tait [9] implemented a WWW meta searching agent, called Hesperus, that clusters web pages based on their similarity to exemplar texts. An exemplar text represents the kind of output that would exemplify a successful search and is found by personal recommendation, or through recommender systems. The agent identifies the lexical chains in a text using Roget's thesaurus. This is used to create an attribute value vector of thesaural categories, called the Generic Document Profile. Using this profile, similarity between a web page retrieved and an exemplar is computed. They experimented their agent initially with two queries and reported that in the case of one query, agent's clustering was significantly correlated with that of human judges. However in the case of a second query, no such correlation could be found.

Vechtomova et al. [43] made use of lexical bonds to quantify lexical cohesion. For each query term, words that co-occur within fixed-size windows identified around each occurrence of the query term in the document are recorded. All of these co-occurring words are then merged to determine the context of the query term in the document. For every pair of query terms, the number of co-occurrences are counted and a lexical cohesion score is obtained. This score is

fused with the BM25 [37] matching function to re-rank the documents. Performance improvements were reported on TREC collections. This way, the authors proved the hypothesis that in a relevant document all query terms are likely to be used in related contexts and tend to share many semantically-related words while in a non-relevant document query terms are less likely to occur in related contexts, and hence they co-occur with fewer common terms. Therefore, it is also shown that relevant documents tend to have a higher level of lexical cohesion between different query terms' contexts than non-relevant documents.

In a recent study, Vechtomova et al. [42] extended their work on lexical bonds. Instead of windows around query terms, they used sentence boundaries. For each sentence containing a query term, they calculate the number of lexical bonds formed between that sentence and other sentences containing different query terms. They experimentally found out that there should exist at least two lexical links between two sentences for them to form a bond. They compute a contribution score for each query term instance using the number of lexical bonds formed by the sentence containing the instance. They sum these contributions and calculate a pseudo-frequency (pf_i) weight for each query term i. Finally they modify and use the BM25 formula (Eqn. 1.8) replacing TF_i with pf_i. They evaluated the performance of their methods on four TREC collections and obtained improvements, though not significant. However, they reported major improvement when they combined this method with a proximity-based method that they also suggest in the same work (described in §2.3).

2.3 Term Proximity in IR

Term proximity-based methods rely on two intuitions: (1) the closer the terms are in a document, the more likely it is that they are related, and (2) the closer the query terms are in a document, the more likely it is that the document is relevant to the query [42].

Some of the proximity-based methods are based on evaluating multi-word

units (*phrases*) in text. These include nominal compounds ('ice cream', 'turn-off valve'), phrasal verbs ('get up', 'run into'), proper nouns ('New York City', 'Albert Einstein') and some idioms ('food for thought', 'nuts and bolts').

Fagan [10] proposed a phrase indexing method controlled by six parameters that incorporate the notion of term specificity and the co-occurrence characteristics of terms into the phrase construction process. The parameters are domain (of co-occurence of phrase elements, like document or sentence), proximity (relative location of phrase elements), df-phrase (document frequency threshold for phrases), df-head (document frequency threshold for phrase heads), df-comp (document frequency threshold for phrase components), and length (the number of elements in a phrase). Retrieval experiments conducted on five document collections revealed that the phrase indexing method performed significantly better than single term indexing for some collections.

Mitra et al. [23] compared the usefullness of phrases recognized using linguistic methods and those recognized by statistical techniques. Statistical phrases were selected as the pairs of non-functional words that occur contiguously in at least 25 documents. The individual words are stemmed and the pair is ordered lexicographically. To identify syntactic phrases, every word in the document is tagged with its part of speech (POS) and certain tags are then recognized as noun phrases. The experiments made on a TREC collection showed that phrases are useful for some queries, the use of phrases does not significantly affect precision at the top ranks, and syntactic phrases perform better than statistical phrases.

Clarke et al. [6] proposed a relevance ranking technique called cover density ranking. Initially the documents are grouped into sets (coordination levels) according to the number of distinct query terms each contains, with the initial ranking of a document based on the set in which it appears. Ranking of documents within a coordination level is based on the proximity and density of query terms within the documents. The cover sets within a document are identified, where a cover refers to the shortest span in the document containing query term instances. The scoring of cover sets is based on two assumptions: (1) the shorter the cover, the more likely the corresponding text is relevant; and (2) the more

covers contained in a document, the more likely the document is relevant. Evaluations made on a TREC test collection demonstrated performance that compares favorably with previous work.

Apart from methods based on capturing phrases, there are also studies aiming to model term dependencies, which are generally ignored by classical IR models as discussed in §1.3.

Metzler and Croft [21] developed a general, formal framework for modeling term dependencies via Markov random fields. They made use of features based on occurrences of single terms, ordered phrases, and unordered phrases. They explored full independence, sequential dependence, and full dependence variants of the model. Ad hoc retrieval experiments were presented on several newswire and web collections and the results showed that significant improvements are possible by modeling dependencies, especially on the larger web collections.

Vechtomova [41] proposed a method of matching and weighting phrases in documents, specifically addressing the problem of weighting overlapping and non-contiguous word sequences in documents. They reported small improvements over a baseline system on a TREC collection.

Rasolofo and Savoy [26] suggested the use of proximity measurement in combination with the BM25 probabilistic model. Their approach is based on the assumption that if a document contains sentences having at least two query terms within them, the probability that this document will be relevant must be greater. Moreover, the closer the query terms are, the higher the relevance probability is. They modified the BM25 weighting scheme so as to consider proximity between query term pairs. They evaluated their approach on three TREC collections and obtained some improvements, though not consistent, on average precision and precision at 5, 10 and 20 documents.

Similarly, Büttcher et al. [5] proposed an integration of term proximity scoring into BM25. Their evaluation on a TREC Terabyte track collection demonstrated better performance on precision at 10 and 20 documents. They also concluded that for stemmed queries the impact of term proximity scoring is larger than for

unstemmed queries.

In the recent study of Vechtomova et al. (also mentioned in §2.2), the authors modify the BM25 weighting function (Eqn. 1.8) replacing TF_i with pf_i where pf_i is the pseudo-frequency of query term i and is computed using its shortest distance to another query term in all sentences it appears. The closer the query terms are, the higher the pseudo-frequency is. They obtained slight improvements by the experiments done on collections.

Chapter 3

System Description

3.1 Overview

As described in §1.4, occurrences of words in text are correlated but classic IR models ignore this, treating words as independent entities. Linguistic theories suggest that the correlation between the words implies the cohesiveness of a text. Lexical cohesion and term proximity are two linguistic properties contributing to cohesiveness. In this work, repetition based lexical cohesion is considered (cf. Table 2.1). Lexical cohesion and term proximity computations are based on collocation (cf. §3.2.2). Figures 3.1 and 3.2 (from [42]) illustrate the formation of proximity based (short-distance) and lexical cohesive (long-distance) relationships in text, respectively.

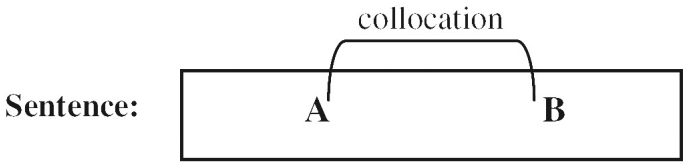


Figure 3.1: Short-distance relationship between query terms

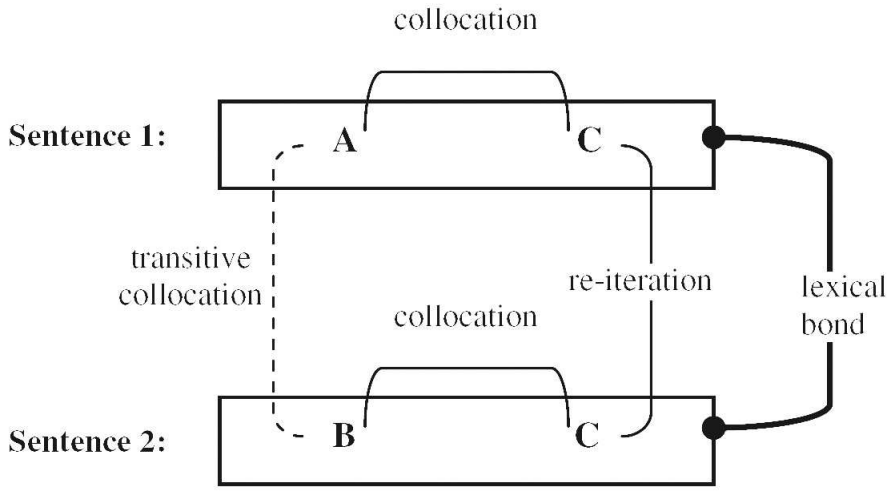


Figure 3.2: Long-distance relationship between query terms

The methodology described in this section aims to detect the degree of cohesiveness between the words using a graph representation where the nodes represent the words and the arcs the strength of cohesion computed based on word co-occurrences. In this way, direct paths between words represent the term proximity and transitive paths represent the lexical cohesion. By exploiting the paths between words, a graph-based cohesion score is obtained.

In order to show the effectiveness of our approach, the graph-based cohesion score is used in an information retrieval task. Performance improvement has already been demonstrated by Vechtomova et al. [43, 42] by means of ranking a document set using lexical cohesion and term proximity between query terms. Similarly, in this thesis the lexical cohesion computed for all query term pairs in each documents is used to re-rank documents in a collection. So, performance improvement in retrieval effectiveness implies that the lexical cohesion and term proximity computations are successful.

3.2 Graph-Based Cohesion Computation

In the following subsections, the basic cohesion computation stages are presented. Subsections §3.2.1 - §3.2.4 describe the steps of computing cohesion score for a document. The final subsection §3.2.5 explains how all the documents in the collection are re-ranked after the cohesion scores are computed for each document.

3.2.1 Document Pre-Processing

The first process applied to a document is tokenizing its content. *Tokenizing* is the process of forming words from the sequence of characters in a document. A simplistic approach would be considering "word" as any sequence of alphanumeric characters of length 3 or more, terminated by a space or other special character. So, for instance, the text:

The company's profit was predicted at $1500.

would produce the following sequence of tokens:

the company profit was predicted at 1500

The next step is the stopping. Words which are too frequent within or among the documents in the collection are not good discriminators. These are called *stopwords*, as the text processing stops when one is seen, and they are thrown out. Throwing out these words decreases index size, increases retrieval efficiency, and generally improves retrieval effectiveness [8]. Articles, prepositions, and conjunctions are natural candidates as stopwords. After stopping, the above sequence of tokens would reduce to:

company profit predicted 1500

After the stopwords are removed, the remaining words are stemmed. A *stem* is the portion of a word which is left after the removal of its affixes (i.e. prefix and suffixes). Stemming reduces the different forms of a word that occur because of inflection (e.g., plurals, tenses) or derivation (e.g., making a verb to a noun by adding the suffix -ation) to a common concept [8]. Applying one of the most popular stemmers, the Porter stemmer [25], to the above tokenized and stopped

text would produce:

compani profit predict 1500

In this work, tokenizing, stopping, and stemming of the documents rely on Okapi IR system's "parse" functionality[1]. Finally, the document is reduced further in order to include in the calculations only the most significant F number of terms determined using the tf-idf weighting scheme (Eqn. 1.6). By this way, only the significant terms which contribute to the actual meaning of the document are kept.

The steps described in the subsequent sections are applied on the tokenized, stopped, and stemmed (i.e. reduced) document, rather than the original full-text document.

3.2.2 Creation of Collocation Matrix

Collocation is defined in various ways by different authors. Hoey's basic definition is adopted in this thesis: *collocation* is the property of language whereby two or more words seem to appear frequently in each other's company [17]. One of these words is called another's *collocate*. Collocation can be systematic or non-systematic. Systematic collocation includes antonyms, members of an ordered set such as [one, two, three], members of an unordered set such as [white, black, red], and part-to-whole relationships like [eyes, mouth, face]. Non-systematic collocation exist between words that tend to occur in similar lexical environments. Words tend to occur in similar lexical environments because they describe things that tend to occur in similar situations or contexts in the world. For instance, the word relationship [garden, digging] is non-systematic [24].

As stated in the previous paragraph, collocation can convey information about the similarity of words' lexical environments. So, it's useful to benefit from collocations while computing cohesion. To find collocations, fixed-sized windows around every instance of each term in the document are identified. A *window* is

[1]http://www.soi.city.ac.uk/ andym/OKAPI-PACK/appendix-j.html#parse

defined as S number of stemmed, non-stopwords to the left and right of a term.

By using the windows identified around each term, the Collocation Matrix (CM) is created for the document. $CM = [m_{ij}]$ is an LxL symmetric matrix where L is the number of distinct terms (i.e. term *types*, not instances) in the reduced document, and each element m_{ij} represents how many times any instance of $term_i$ occurs in the same window (i.e., collocates) with any instance of $term_j$.

3.2.3 Conversion of CM into Cohesion Graph

An undirected, weighted Cohesion Graph, $CG = (N, A)$ is created from the CM such that;

$N = \{\text{term types in the document}\}$, and
$A = \{(i, j) : w_{ij} = \text{collocation strength between } term_i \text{ and } term_j\}$.

To calculate the collocation strength between terms, the co-occurrence frequencies, i.e. m_{ij} values, from the CM are used. So for an arc $(i, j) \in A, w_{ij} = m_{ij}$.

In CG, a direct path between two nodes implies that the two terms represented by these nodes co-occur in the same window at least once (term proximity). A multi-hop path implies that the two terms are related transitively by means of some other common term(s) (lexical cohesion). It is assumed that, as these terms co-occur within a common subset of terms, they should also be contextually related.

3.2.4 Calculation of Cohesion Graph Score

The Cohesion Graph Score (CGS) of query terms for a document is derived from the strength of the paths between query terms. The algorithm to calculate the score of a document $\{d\}$ for a query term set $\{query\_term\_set\}$ is as follows:

begin
  $\{query\_terms\} = \{d\} \cap \{query\_term\_set\}$;
  if $|\ \{query\_terms\}\ | < 2$ **then**
    return *0*;
  else
    foreach *query term pair* $(q_i, q_j) : q_i, q_j \in \{query\_term\_set\}$ **do**
      construct P, set of paths between q_i & q_j with max length of M;
      foreach *path* $p_k \in P$ **do**
        calculate path score $PATH\_SC(q_i, q_j)_k$;
      end
      calculate pair score $PAIR\_SC(q_i, q_j)$ using $PATH\_SC(q_i, q_j)_k$;
    end
    calculate document score DOC_SC using $PAIR\_SC(q_i, q_j)$;
    return DOC_SC; `// DOC_SC = CGS`
  end
end

Algorithm 1: Algorithm to calculate CGS

As the algorithm describes, there are three levels of computation to reach CGS: path level, query term pair level, and document level. Separation of computations allows investigating cohesion characteristics at different levels. For each level there are a number of alternative methods of calculation. These are explained below and summarized in Table 3.1.

DOC_SC (CGS)		**PAIR_SC**		**PATH_SC**	
Method	**Symbol**	**Method**	**Symbol**	**Method**	**Symbol**
Average	Av	Average	Av	Average	Av
Multiplication	Ml	Minimum	Mn	Minimum	Mn
Sum	Sm	Maximum	Mx	Maximum	Mx
		Multiplication	Ml		
		Sum	Sm		

Table 3.1: Alternative methods to calculate path, pair and document scores

3.2.4.1 Calculation of the Path Score (PATH_SC)

The following methods were chosen to compute the path score:

- taking the average of the weights of the arcs in the path (Av)
- taking the maximum weighted arc in the path (Mx)
- taking the minimum weighted arc in the path (Mn)

The minimum and maximum values identify the weakest and strongest chains in the path. Averaging assumes that the overall path strength lies somewhere between these extreme values. Trivially, any of the path score calculation methods described above reduces to the same value for direct links (without any intermediate node).

3.2.4.2 Calculation of the Pair Score (PAIR_SC)

Usually there are several paths between query term pairs. The score of a query term pair is computed by one of the following methods:

- taking the average of path scores (Av)
- taking the maximum path score (Mx)
- taking the minimum path score (Mn)
- taking the product of path scores (Ml)
- taking the sum of path scores (Sm)

Summation, multiplication and averaging of path scores are chosen in order to investigate the effect of the number of distinct paths between query term pairs. To save from computation, multiplication is implemented as summation of the logarithms of path scores.

3.2.4.3 Calculation of the Document Score (DOC_SC, CGS)

The final score of the document is reached by either:

- summing all pair scores (Sm), or
- multiplying all pair scores (Ml)

The latter method is useful in penalizing documents where one or more of the query term pairs are weakly linked. While executing this method, a non-existing query term yields a pair score of y, $0 \leq y \leq 1$, with the other query. $y = 0$ means that the document will get a CGS of 0 if at least one query term is missing in it. $y = 1$ means that non-existence of a query term in a document will not affect its CGS at all. A value in between penalizes the document for missing query terms but prevents it from being treated as a document containing none of the query terms.

3.2.5 Re-Ranking of Documents

To understand its reliability, CGS is used in re-ranking the documents of a collection in response to a set of queries. The queries are tokenized, stopped, and stemmed using Okapi's "parse" functionality, as done in reducing documents. Using the resulting query terms, CGS is calculated as described in steps §3.2.1 - §3.2.4.

Documents are re-ranked either directly by their CGS scores or by fusing this score with their BM25 (Eqn. 1.8) scores. The fused score, COMB-CGS, for a document is calculated as follows:

$$COMB - CGS = MS + x \cdot CGS \tag{3.1}$$

where MS is the matching score (BM25) returned by Okapi IR system and x is a tuning constant to regulate the final score.

3.3 Improving CGS

In order to improve the performance of the basic CGS method, the modifications described in the following subsections were applied at different steps of calculation.

3.3.1 Consideration of document length

CGS calculation does not take into account the length of documents in the collection. A long and a short document giving exactly the same CGS may experience a bias in favor of the long document because a long document is expected to score much, due to the higher number of collocations it should contain. A short document with the same score should show that it is more cohesive than a longer document. To normalize the score, a variant method, CGS_{DL}, is built where the weight of the arc, w_{ij}, between each node pair (i, j) is updated as follows:

$$w_{ij} = m_{ij} \cdot \ln\left(\frac{AVDL}{DL} + 1\right) \tag{3.2}$$

where DL is the length of document, $AVDL$ is the average document length of the retrieved set per query, and m_{ij} is the co-occurrence frequency of terms i and j. In this way, a long document is penalized for its length whilst a shorter one is rewarded.

3.3.2 Consideration of inverse document frequency

In the basic CGS method, solely intra-document relationships (i.e. co-occurrence frequencies within document) between the terms are considered. During preprocessing *idf* weights of terms are used to reduce document but during cohesion computation there is no use of any collection-wide term information. To include the collection distribution of terms in CG, a new method, CGS_{IDF}, is developed where the weight of the arc, w_{ij}, between each node pair (i, j), is updated as

follows:

$$w_{ij} = m_{ij} \cdot f(idf_i, idf_j) \tag{3.3}$$

where m_{ij} is the co-occurrence frequency of terms i and j. The function $f(idf_i, idf_j)$ returns a value based on the idf weights of terms by one of the following methods:

- taking the average of idf weights (Av)
- taking the maximum idf weight (Mx)
- taking the minimum idf weight (Mn)
- taking the sum of idf weights (Sm)

Once the graph is updated, CGS_{IDF} is calculated as described in §3.2.4.

3.3.3 Incorporating BM25 matching function

In the COMB-CGS method (Eqn. 3.1), CGS is fused with BM25. CGS and BM25 are two complementary methods, the former considering intra-document lexical cohesive relationships and the latter collection-wide term statistics. Instead of fusing, another possibility is to incorprate BM25 into CGS. This is done by a new variant method, CGS_{TW}, in which CG arc weights are updated as follows:

$$w_{ij} = m_{ij} \cdot g(TW_i, TW_j) \tag{3.4}$$

where m_{ij} is the co-occurrence frequency of terms i and j. BM25 term weights, TW_i and TW_j, are computed according to Eqn.1.8. The function $g(TW_i, TW_j)$ returns a value using one of the following methods:

- taking the average of BM25 weights (Av)
- taking the maximum BM25 weight (Mx)

- taking the minimum BM25 weight (Mn)
- taking the sum of BM25 weights (Sm)

After the graph is updated, CGS_{TW} is calculated as described in §3.2.4.

Chapter 4

Experimental Design

4.1 Procedure

In order to show the effectiveness of CGS, information retrieval experiments were conducted based on TREC test collections (§4.3). Short queries were created from all non-stopword terms in the "Title" fields of TREC topics (Figure B.9). Single-term queries were not considered since CGS requires at least two query terms to be computed. Top T documents are retrieved using Okapi IR System and then re-ranked by CGS and COMB-CGS methods. Okapi is briefly described in §4.2. Fixed and tested parameters are provided in §4.4.

The retrieval performance of the methods implemented were evaluated using trec-eval[1], which is a standard program written by Chris Buckley for scoring the quality of a retrieval result. Trec-eval provides a common implementation for over 100 different evaluation measures that ensures issues such as interpolation are handled consistently. Figure B.10 shows a sample output generated by trec-eval. Despite large number of available evaluation measures, a much smaller set of measures has emerged as the de facto standard by which retrieval effectiveness is characterized. These measures include the recall-precision (R-Prec) graph, mean average precision (MAP) and precision at ten retrieved documents (P10) [44].

[1] http://trec.nist.gov/act_ part/tools.html

These three major metrics are used in this work to evaluate and compare the developed methods.

4.2 Okapi IR System

Okapi[2] is an experimental text retrieval system based at City University, London. It started as an online library catalogue system and has since been made available to groups of researchers. The structure of the Okapi mainly consists of the following components: indexing routines, search engine (Basic Search System or BSS), and various interface systems [27].

The Okapi team at City University has taken part in every round of TREC, which, as stated in [29], has encouraged and made possible substantial developments both in system design and in underlying models. However, it is also noted that BM25 formula (Eqn. 1.8) has remained more-or-less fixed since TREC-3.

Okapi povides three types of stemming: weak, strong, and none. It parses and indexes documents according to a GSL file which is a list of stop terms, stop marks, phrases and synonym groups. During indexing, the GSL file can be tailored for a collection.

4.3 Collections

The following standard test collections were used during experiments:

1. TREC 2003 HARD track collection (*HARD03*): 372,219 documents from 3 newswire corpora and U.S. government documents. Two of the 50 topics had no relevant documents and were excluded from the official HARD 2003 evaluation [1]. Two more topics were single-term queries so were also excluded (See Figure B.11).

[2]http://www.soi.city.ac.uk/~andym/OKAPI-PACK

2. TREC 2004 HARD track collection (*HARD04*): 652,710 documents from 8 newswire corpora and 50 topics. Five of the topics had no relevant documents and were excluded from the official HARD 2004 evaluation [2]. Five more topics were single-term queries so were also excluded (See Figure B.12).

3. TREC 2005 HARD track collection (*HARD05*): 1,033,461 documents from 3 newswire corpora and 50 topics [3]. One of the topics was a single-term query and was excluded (See Figure B.13).

Instead of separating the collections for testing and training, in the next chapter, the best run in one collection is presented in the other as well. In this way it is possible to cross-validate the evaluation results.

4.4 Parameters

The parameters described in §4.4.1 were fixed throughout all experiments. The variable parameters that are tested are given in §4.4.2 with values tried.

4.4.1 Fixed Parameters

In BM25 equation (Eqn. 1.8):

- $k_1 = 1.2$
- $b = 0.75$
- $r = R = 0$ - no prior relevance judgements

In CGS calculation (§3.2):

- $T = 1000$ - number of documents retrieved
- $P = 2$ (i.e. one intermediate node) - maximum hop count for a path

4.4.2 Variable Parameters

In CGS calculation (§3.2):

- F = 50, 100, 1000 - number of terms considered
- S = 5, 10, 15 - window size
- y = 0.0, 0.2, 0.5, 0.8, 1.0 - contribution of a non-existing query term to multiplication during DOC_SC computation

In COMB-CGS computation (Eqn. 3.1):

- x = 0.008, 0.01, 0.125, 0.25, 0.5, 1.0, 2.0 - tuning constant

Chapter 5

Evaluation Results

5.1 Performance Comparison of Methods

Table 5.1 summarizes the performance of CGS and COMB-CGS against the benchmark, Okapi BM25. Improvements significant at 0.05 by two-tailed paired t-test are marked by *. The table reveals that CGS performs significantly better only at P10 on HARD05. COMB-CGS outperforms BM25 at all metrics, significantly on HARD04 and HARD05. It also performs better than CGS on all collections and metrics.

METHOD	HARD03			HARD04			HARD05		
	MAP	P10	RPREC	MAP	P10	RPREC	MAP	P10	RPREC
BM25	0.3258	0.5478	0.3464	0.2014	0.3025	0.2317	0.1697	0.3694	0.2307
CGS	0.2524	0.4435	0.2857	0.1872	0.3450	0.2447	0.1747	0.4490 *	0.2347
COMB-CGS	0.3281	0.5783	0.3546	0.2311 *	0.3825 *	0.2749 *	0.1975 *	0.4612 *	0.2587 *

Table 5.1: The highest performance scores of BM25, CGS and COMB-CGS

The individual retrieval performances of CGS and COMB-CGS for each topic of every collection are shown in figures B.1 - B.6.

As described previously, CGS is calculated using solely intra-document relationships between terms. Therefore, it does not contain any collection-wide term information. This is probably why CGS on its own does not always produce results as good as the baseline Okapi BM25 system. However, when the scores of

both systems are fused (Eqn. 3.1), the results are better than the either system on its own, suggesting that BM25 and CGS capture complementary relevance information.

5.2 Parameter Analysis of CGS

The performance of CGS on three datasets and three metrics is summarized in Table 5.2. The following parameters are displayed: window size (S), number of terms (F) used in document representations, and the methods used in calculating path, pair and document scores (Av, Ml, Mn, Mx, Sm). The highest scores for a given collection-evaluation measure combination are typed in bold.

Best combinations found			**Sets and metrics tested on**								
F	**S**	**Method**	**HARD03**			**HARD04**			**HARD05**		
			MAP	**P10**	**RPREC**	**MAP**	**P10**	**RPREC**	**MAP**	**P10**	**RPREC**
1000	**15**	**Ml-Sm-Mn**	**0.2524**	0.4326	**0.2857**	0.1792	0.3275	0.2346	0.1739	0.4286	0.2341
100	**15**	**Ml-Sm-Av**	0.2367	**0.4435**	0.2646	0.1791	**0.3450**	0.2214	0.1605	0.4469	0.2129
1000	**15**	**Ml-Sm-Av**	0.2485	0.4152	0.2850	**0.1872**	0.3225	0.2417	0.1711	0.4245	0.2323
1000	**10**	**Ml-Sm-Av**	0.2455	0.4087	0.2767	0.1868	0.3375	**0.2447**	0.1732	0.4204	0.2340
1000	**5**	**Ml-Sm-Mn**	0.2396	0.4326	0.2789	0.1814	0.3250	0.2310	**0.1747**	0.4163	0.2323
100	**15**	**Ml-Ml-Av**	0.2349	0.4283	0.2641	0.1807	0.3300	0.2255	0.1627	**0.4490**	0.2124
1000	**10**	**Ml-Sm-Mn**	0.2456	0.4130	0.2828	0.1811	**0.3450**	0.2378	0.1736	0.4163	**0.2347**

Table 5.2: Best performing runs for CGS

There is no best run with $F = 50$. $F = 1000$ yields the best results in MAP and R-PREC, while $F = 100$ gives the best result in P10 on all collections. This suggests that it is best to represent the documents (F) with more terms for good performance in general, but with fewer terms for high precision (e.g. P10).

For window size, $S = 15$ is the most popular value, followed by $S = 10$ at R-PREC on HARD04 and HARD05, and by $S = 5$ at MAP on HARD05. But it is observed in Table 5.3 that $S = 15$ performs either the best or nearly the best (for the same fixed window size and method combinations) in all collections and metrics. Therefore, it can be understood that keeping windows larger (i.e. considering longer collocation distances) is better.

In calculating the document score, multiplying (Ml) the pair scores performs better than summing (Sm) them. The superiority of multiplication over summing

Best combinations		Sets and metrics tested on								
F	Method	HARD03			HARD04			HARD05		
		MAP	P10	RPREC	MAP	P10	RPREC	MAP	P10	RPREC
1000	**Ml-Sm-Mn**	**0.2524**	0.4326	**0.2857**	0.1792	0.3275	0.2346	0.1739	0.4286	0.2341
100	**Ml-Sm-Av**	0.2367	**0.4435**	0.2646	0.1791	**0.3450**	0.2214	0.1605	0.4469	0.2129
1000	**Ml-Sm-Av**	0.2485	0.4152	0.2850	**0.1872**	0.3225	0.2417	0.1711	0.4245	0.2323
1000	**Ml-Sm-Av**	0.2485	0.4152	0.2845	0.1872	0.3225	**0.2417**	0.1711	0.4245	0.2323
1000	**Ml-Sm-Mn**	0.2524	0.4326	0.2857	0.1792	0.3275	0.2346	**0.1739**	0.4286	**0.2341**
100	**Ml-Ml-Av**	0.2349	0.4283	0.2641	0.1807	0.3300	0.2255	0.1627	**0.4490**	0.2124

Table 5.3: CGS runs for S=15

suggests that the more the pair scores vary across query term pairs in a document the less the document is cohesive with respect to query terms, hence, the less likely that the document is relevant. Thus, in relevant documents there are higher number of query term pairs that are lexically connected, and the strength of this connection tends to be uniform among all query term pairs.

It can also be observed from the results that summing (Sm) path scores to arrive pair scores is better than taking the minimum (Mn), maximum (Mx), multiplication (Ml) or average (Av) of the path scores, except for P10 at HARD05. For the same F and S values and document and path score calculations methods, Sm performs comparable to Ml for P10 at HARD05 though (see Table 5.4). This result indicates that the higher the number of distinct paths between a query term pair the more likely the document is relevant. Thus, in relevant documents query terms tend to have more common collocates than in non-relevant documents.

F	S	Method	MAP	P10	RPREC
100	**15**	**Ml-Ml-Av**	0.1627	**0.4490**	0.2124
100	**15**	**Ml-Sm-Av**	0.1605	0.4469	0.2129

Table 5.4: Ml vs. Sm as pair scores for F=100 S=15 in HARD05

In obtaining the path scores averaging the weights of the arcs (Av), followed by taking the minimum of arc weights (Mn) are the two best performing methods. Averaging is preferred for high precision (P10) while a worst-case value of taking the minimum arc (i.e. terms connected as weakest) sometimes performs better for MAP and R-PREC.

Set	HARD03			HARD04			HARD05		
Metric	MAP	P10	RPREC	MAP	P10	RPREC	MAP	P10	RPREC
y	0.8	0.2	1.0	1.0	0.2,0.5,0.8,1.0	0.2	0.8	0.8,1.0	0.5

Table 5.5: Best performing y values for CGS

As explained in §3.2.4.3, during document score calculation, a non-existing query term contibutes a value of y in multiplying (Ml) pair scores. Ml is always the best document score calculation method and y values providing the best score are given in Table §5.5. The table shows that there is no agreement on a particular y value within or across neither a collection nor a metric. However, deeper observation of performance results reveals that all values, except for $y = 0.0$ which performs much worst, achieve very close performance within the same F, S, and score calculation methods.

5.3 Parameter Analysis of COMB-CGS

The best performing COMB-CGS runs are given in Table 5.6 for three collections (the highest scores for a given collection-evaluation measure combination are typed in bold). The table shows that there is no unique combination of parameters that yields the highest score in all measures in a collection or for a given evaluation metric on three collections. $F = 1000$ always performs best for R-PREC and $x = 0.125$ is the best tuning constant value for document scores calculated using multiplication. Comparison of Table 5.2 and Table 5.6 suggests that the selection of parameters and methods depends on the document collection more in COMB-CGS runs than in CGS runs.

Best combinations found				**Sets and metrics tested on**								
F	**S**	**x**	**Method**	**HARD03**			**HARD04**			**HARD05**		
				MAP	**P10**	**RPREC**	**MAP**	**P10**	**RPREC**	**MAP**	**P10**	**RPREC**
50	**15**	**0.125**	**Ml-Mn-Mx**	**0.3281**	0.5652	0.3475	0.2108	0.2950	0.2443	0.1694	0.3755	0.2340
50	**5**	**0.125**	**Ml-Ml-Av**	0.3269	**0.5783**	0.3477	0.2227	0.3350	0.2564	0.1781	0.4143	0.2368
1000	**5**	**0.25**	**Sm-Mn-Av**	0.3243	0.5370	**0.3546**	0.2172	0.2975	0.2475	0.1675	0.3490	0.2316
100	**15**	**0.25**	**Sm-Mx-Av**	0.2879	0.4717	0.3310	**0.2311**	0.3575	0.2698	0.1698	0.3939	0.2302
50	**5**	**0.5**	**Sm-Ml-Mx**	0.2954	0.4870	0.3313	0.2260	**0.3825**	0.2623	0.1792	0.3959	0.2348
1000	**15**	**0.125**	**Ml-Sm-Av**	0.2916	0.4587	0.3312	0.2245	0.3500	**0.2749**	0.1851	0.4367	0.2503
1000	**5**	**0.5**	**Sm-Sm-Mn**	0.2652	0.3717	0.3130	0.2170	0.3425	0.2511	**0.1975**	0.4367	0.2576
100	**15**	**0.125**	**Ml-Ml-Av**	0.3182	0.5174	0.3484	0.2260	0.3575	0.2706	0.1824	**0.4612**	0.2422
1000	**5**	**0.5**	**Sm-Sm-Av**	0.2535	0.3478	0.3085	0.2188	0.3450	0.2569	0.1911	0.4204	**0.2587**

Table 5.6: Best performing runs for COMB-CGS

As in CGS, there is not a common y value within or across a collection or a metric in COMB-CGS (see Table 5.7).

Collection	HARD03			HARD04			HARD05		
Metric	MAP	P10	RPREC	MAP	P10	RPREC	MAP	P10	RPREC
y	0.5	0.2	-	-	-	0.2	-	0.5, 1.0	-

Table 5.7: Best performing y values for COMB-CGS

5.4 Impact of Variant Methods of CGS

Of the three modifications suggested for improving the performance of CGS, consideration of document length (§3.3.1) resulted in performance improvement only in the HARD03 collection (See Table 5.8). This was expected because this collection, consisting of a heterogeneous set of news and government documents with highly varying document lengths, distinguishes itself from the other two collections which contain solely news articles of usually close lengths.

METHOD	HARD03		
	MAP	P10	R-PREC
CGS	0.2524	0.4435	0.2857
CGS_{DL}	**0.2785**	**0.4870**	**0.3066**

Table 5.8: HARD03 performance with consideration of document length

In all collections, considering inverse document frequency (§3.3.2) improved P10 performance. Table 5.9 provides the numbers together with the best performing $f(idf_i, idf_j)$ methods in parantheses.

METHOD	HARD03 P10	HARD04 P10	HARD05 P10
CGS	0.4435	0.3450	0.4490
CGS_{IDF}	**0.4652 (Mn)**	**0.3550 (Av/Sm)**	**0.4673 (Mn)**

Table 5.9: P10 improvement with consideration of IDF

Incorporating BM25 matching function (§3.3.3) resulted in improvement in MAP and R-Prec (except for HARD04) in all collections. The results are given together with the best performing $g(TW_i, TW_j)$ methods in parantheses in Table 5.10.

METHOD	HARD03		HARD04		HARD05	
	MAP	RPREC	MAP	RPREC	MAP	RPREC
CGS	0.2524	0.2857	0.1872	**0.2447**	0.1747	0.2347
CGS_{TW}	**0.2574 (Mx)**	**0.2947 (Mn)**	**0.1951 (Mx)**	0.2430 (Various)	**0.1792 (Sm)**	**0.2409 (Mn)**

Table 5.10: MAP and R-PREC improvement with BM25 incorporation

Chapter 6

Conclusion

Context-awareness is a crucial concern in information retrieval. A document and a query having matching words does not necessarily imply that the document is relevant to the query. The words may reside in the same document but may not share a common context. As opposed to traditional "bag-of-words" retrieval methods, by adapting linguistic theories and incorporate language processing techniques into IR tasks it is possible to perform contextual information retrieval. In this work different methods for document ranking based on lexical cohesion and proximity among query terms in a document were investigated. To compute the degree of cohesion in a document with respect to a query, a document is interpreted as a graph whose nodes are the terms in the document, and arcs representing the strength of association between the terms connected by it. The associations a term has with other terms in the cohesion graph constitute its context in the document. The overall strength of the cohesive relationships between all query terms in a document is indicator of a common context that makes the document relevant to a given query.

6.1 Novelty and Implications of this Study

This study extends the previous studies on cohesion based IR with the following major contributions:

- a new graph representation for documents which is created by the use of collocation information and able to capture both term proximity and lexical cohesion in text,
- two new matching functions based on cohesion in document aiming to improve retrieval effectiveness,
- several linguistic implications about the characteristics of cohesion stemming from score calculation methods and parameters.

The experiments made revealed the following implications:

- BM25 and CGS capture complementary relevance information,
- representing the documents with more terms provides good performance,
- larger windows (i.e. considering longer collocation distances) increases performance,
- in relevant documents there are higher number of query term pairs that are lexically connected, and the strength of this connection tends to be uniform among all query term pairs,
- in relevant documents query terms tend to have more common collocates than in non-relevant documents,
- normalizing cohesion score by average document length is useful in improving retrieval performance on heterogeneous collections consisting of documents with varying lengths,
- considering *idf* weights improves P10,

- incorporating BM25 function into CGS improves MAP and R-PREC.

The methodology described in this work can be used in:

- improving an IR system: CGS can be incorporated and fused with the underlying matching functions of an existing text retrieval engine improving effectiveness,
- summarizing documents: by analyzing the links between terms and consider terms' distribution frequencies within document or collection, important terms can be determined and used in summarizing a document,
- query expansion and re-formulation: using the links query terms form with non-query terms, highly collocated terms can be included in a revised query,
- understanding cohesion characteristics in a collection: by analyzing the best performing document, pair, and path calculations methods, impression about cohesion in different types of collections can be obtained,
- visualizing documents: by enriching the graph representation with visual cues, an interface to preview documents can be provided to users. Enhancement ideas include, but are not limited to, highlighting query term nodes, re-sizing nodes by their weights, and thickening arcs proportional to collocation frequency. This way, a user can evaluate the relevance of a document without reading the whole document.

6.2 Further Research Directions

This study can be extended in several directions:

- Instead of fixed-sized windows, lexical structures, like sentences or paragraphs, may be used as boundary for windows,
- In addition to repetition, other lexical cohesive relationships like synonymy, hypernymy, and etc. may be used in determining collocations,

- Standard graph-based methods may be used to calculate CGS. For instance, collocations may be determined by selecting k-nearest neighbors and the score between two query terms can be determined by random walk,
- Different merging algorithms and matching functions may be used in calculating COMB-CGS, instead of a linear combination with BM25,
- The performance of the methods suggested may be tested on corpora other than news collections like blogs, web documents, and etc.

Bibliography

[1] J. Allan. HARD track overview in TREC 2003: High accuracy retrieval from documents. In *Proceedings of TREC 2003*, pages 24–37, 2004.

[2] J. Allan. HARD track overview in TREC 2004: High accuracy retrieval from documents. In *Proceedings of TREC 2004*, pages 25–35, 2005.

[3] J. Allan. HARD track overview in TREC 2005: High accuracy retrieval from documents. In *Proceedings of TREC 2005*, pages 52–68, 2006.

[4] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, NY, USA, 1999.

[5] S. Büttcher, C. L. A. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. In *SIGIR'06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 621–622, 2006.

[6] C. L. A. Clarke, G. V. Cormack, and E. A. Tudhope. Relevance ranking for one to three term queries. *Information Processing and Management*, 36(2):291–311, 2000.

[7] B. Croft, D. Metzler, and T. Strohman. Evaluating search engines. Online at http://www.pearsonhighered.com/croft1epreview/pdf/chap8.pdf, 2008.

[8] B. Croft, D. Metzler, and T. Strohman. Processing text. Online at http://www.pearsonhighered.com/croft1epreview/pdf/chap4.pdf, 2008.

[9] J. Ellman and J. Tait. Meta searching the web using exemplar texts: Initial results. In *Proceedings of the 20th Annual Colloquium on IR Research*. British Computer Society's Information Retrieval Specialist Group (BCS-IRSG), 1998.

[10] J. L. Fagan. The effectiveness of a nonsyntatic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science*, 40(2):115–132, 1989.

[11] E. Greengrass. Information retrieval: A survey. Online at http://www.csee.umbc.edu/cadip/readings/IR.report.120600.book.pdf, 2000.

[12] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[13] M. A. K. Halliday and R. Hasan. *Cohesion in English*. Longman, London, UK, 1976.

[14] M. A. K. Halliday and R. Hasan. *Language, Context, and Text: Aspects of Language in a Social-Semiotic Perspective*. Oxford University Press, Oxford, UK, 1989.

[15] R. Hasan. Coherence and cohesive harmony. In *Understanding Reading Comprehension*, pages 181–219. International Reading Association, Newark, DE, USA, 1984.

[16] M. Hoey. *Patterns of Lexis in Text*. Oxford University Press, Oxford, UK, 1991.

[17] M. Hoey. *Lexical Priming: A new theory of words and language*. Routledge, Abingdon, UK, 2005.

[18] P. Ingwersen. *Information Retrieval Interaction*. Taylor Graham, London, UK, 1992.

[19] C. Küçükkeçeci. Chisio: A visual framework for compound graph editing and layout. Master's thesis, Bilkent University, June 2007.

[20] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[21] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *SIGIR'05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 472–479, 2005.

[22] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[23] M. Mitra, C. Buckley, A. Singhal, and C. Cardie. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO'97*, pages 200–214, 1997.

[24] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

[25] M. F. Porter. *Readings in Information Retrieval*, chapter An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[26] Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. In *Proceedings of the 25th European Conference on Information Retrieval Research (ECIR 2003)*, pages 207–218, 2003.

[27] S. E. Robertson. Overview of the Okapi projects. *Journal of Documentation*, 53(1):3–7, 1997.

[28] S. E. Robertson and K. Spärk Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[29] S. E. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life: Okapi at TREC. *Information Processing and Management*, 36(1):95–108, 2000.

[30] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *The First Text REtrieval Conference (TREC-1)*, pages 21–30. NIST, Department of Commerce, 1993.

[31] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-2. In *The Second Text REtrieval Conference (TREC-2)*, pages 21–34. NIST, Department of Commerce, 1994.

[32] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *The Third Text REtrieval Conference (TREC-3)*, pages 109–126. NIST, Department of Commerce, 1995.

[33] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[34] G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of ACM*, 26(12):1022–1036, 1983.

[35] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1983.

[36] K. Spärk Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[37] K. Spärk Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and status. Technical report, Computer Laboratory, University of Cambridge, August 1998.

[38] M. A. Stairmand. Textual context analysis for information retrieval. In *SIGIR'97: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 140–147. ACM, 1997.

[39] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, UK, 2nd edition, 1979.

[40] C. J. van Rijsbergen. The science of information retrieval: its methodology and logic. In *Conferentie Informatiewetenschap in Nederland*, pages 21–38. RABIN, 1990.

[41] O. Vechtomova. Noun phrases in interactive query expansion and document ranking. *Information Retrieval*, 9(4):399–420, 2006.

[42] O. Vechtomova and M. Karamuftuoglu. Lexical cohesion and term proximity in document ranking. *Information Processing and Management*, 44(4):1485–1502, 2008.

[43] O. Vechtomova, M. Karamuftuoglu, and S. E. Robertson. On document relevance and lexical cohesion between query terms. *Information Processing and Management*, 42(5):1230–1247, 2006.

[44] E. M. Voorhees. TREC: Improving information access through evaluation. *Bulletin of the American Society for Information Science and Technology*, 32(1):16–21, 2005.

[45] E. M. Voorhees and D. K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press, Cambridge, MA, USA, 2005.

Appendix A

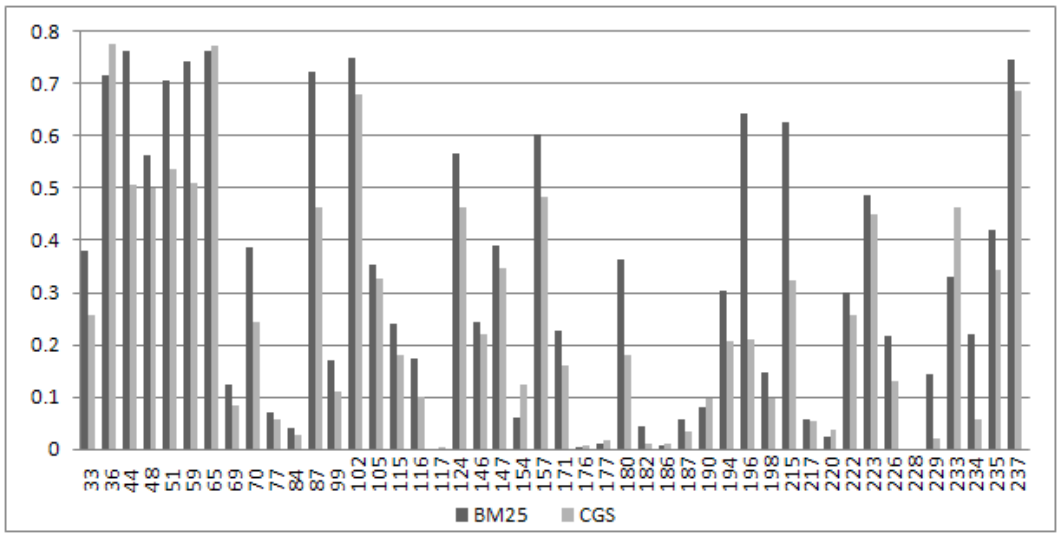
Tables

	Lexical cohesion	Reference	Conjunction	Ellipsis	Substitution
Children's fiction	7	8	3	0	0
Oral narrative	26	12	11	6	2
Sonnet	13	8	2	0	0
Autobiography	20	10	1	0	1
Dramatic dialogue	9	13	5	12	4
Reported interview	17	20	4	1	1
Transcribed interview	15	10	4	7	2
	107	81	30	26	10
(including Conjunction)	42%	32%	12%	10%	4%
(excluding Conjunction)	48%	36%		12%	4%

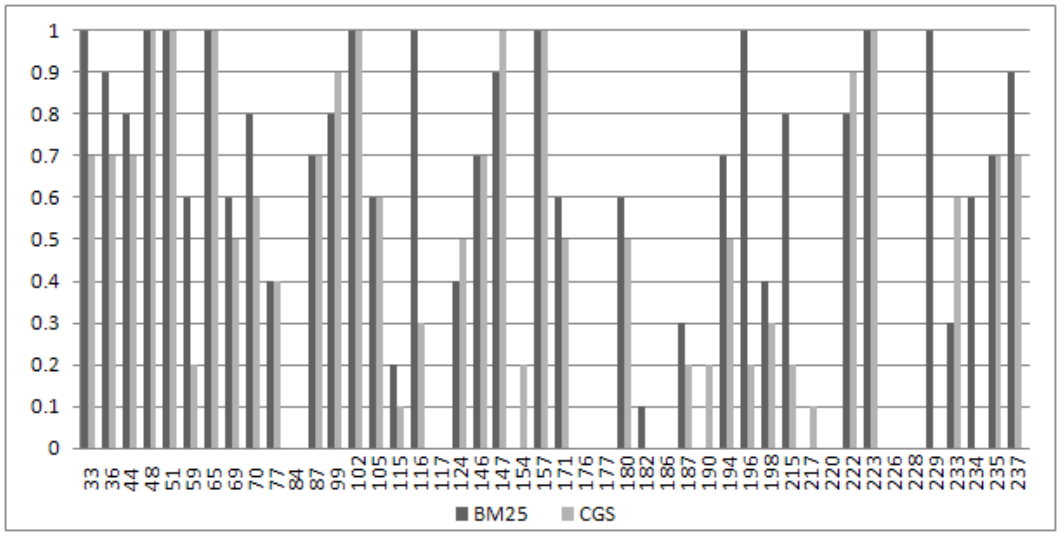
Table A.1: Distribution of classes of cohesive ties for different kinds of texts

Appendix B

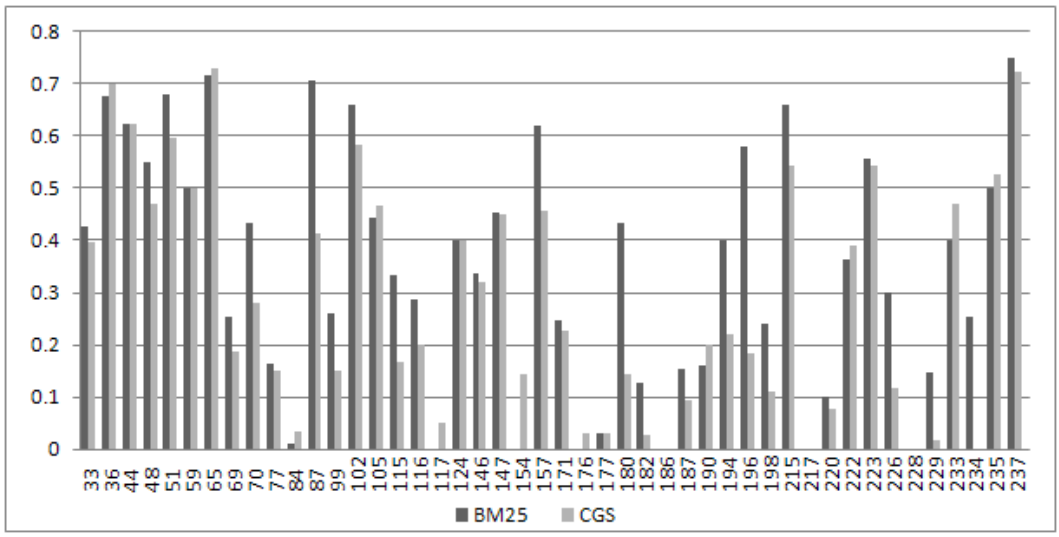
Figures



(a) Topic number vs. MAP

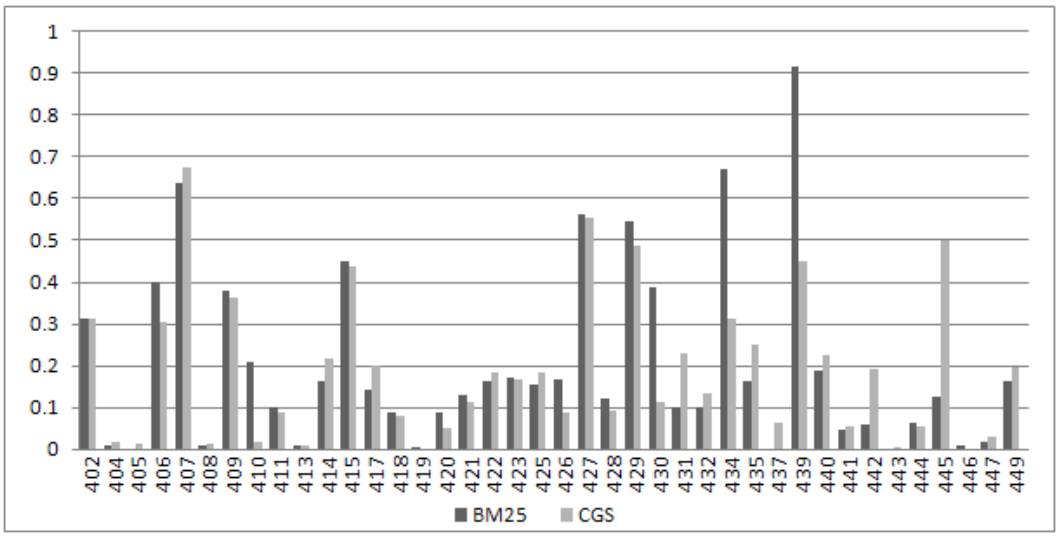


(b) Topic number vs. P10

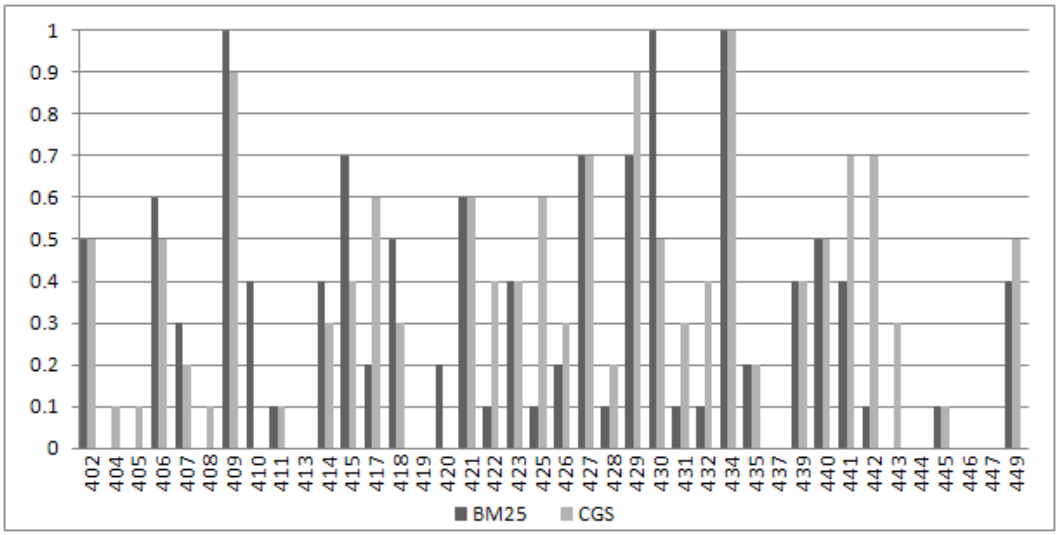


(c) Topic number vs. R-PREC

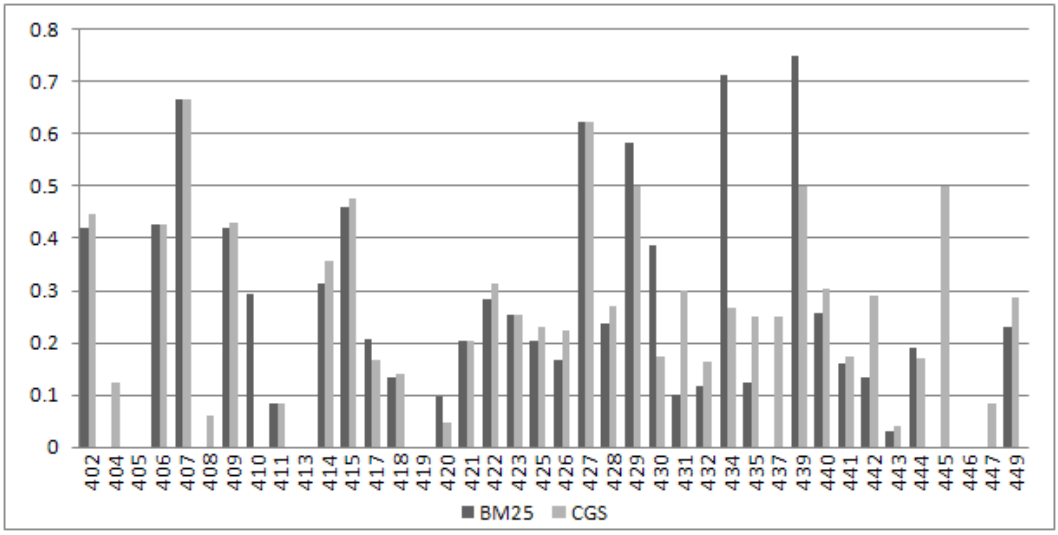
Figure B.1: Query-by-query retrieval performance of CGS on HARD03



(a) Topic number vs. MAP

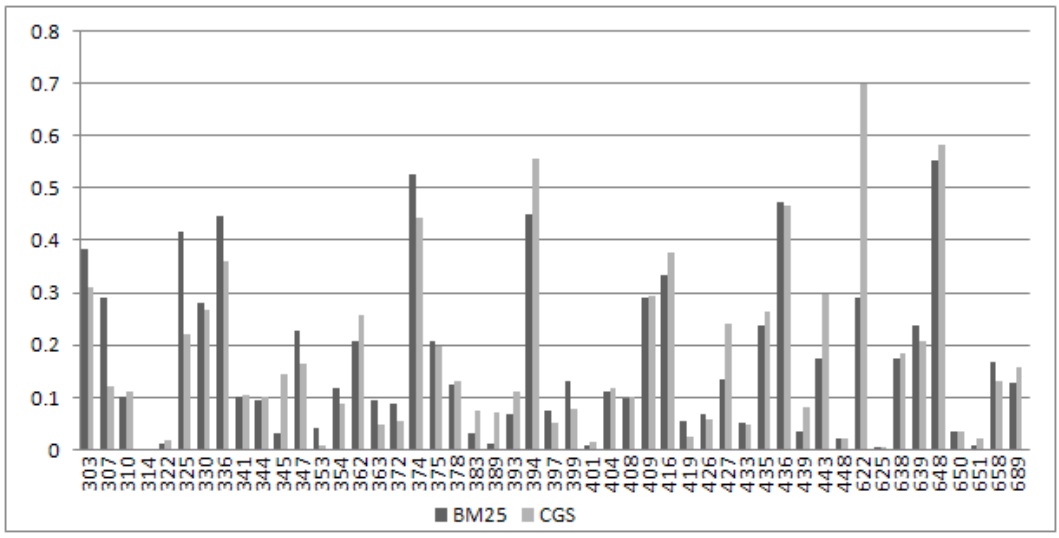


(b) Topic number vs. P10

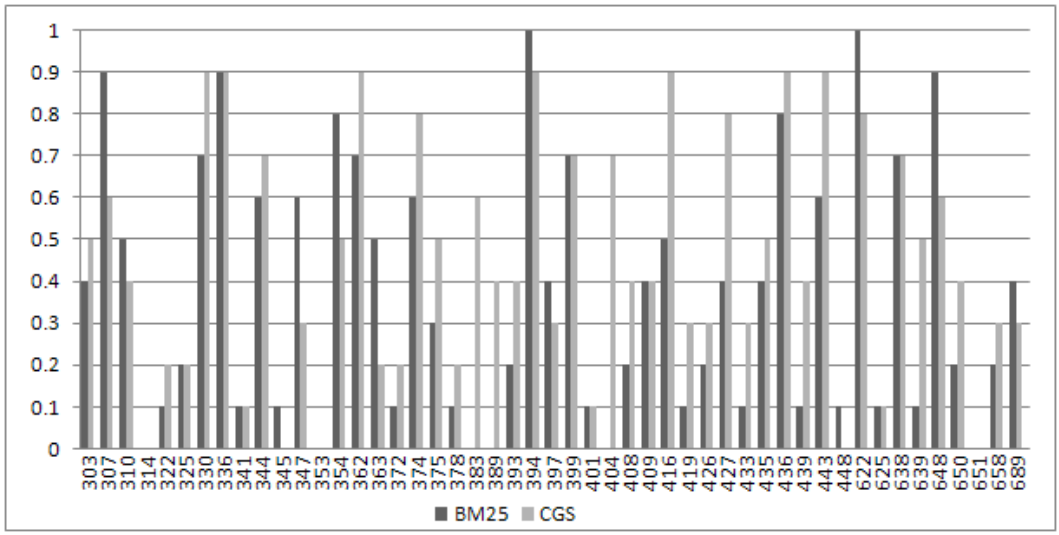


(c) Topic number vs. R-PREC

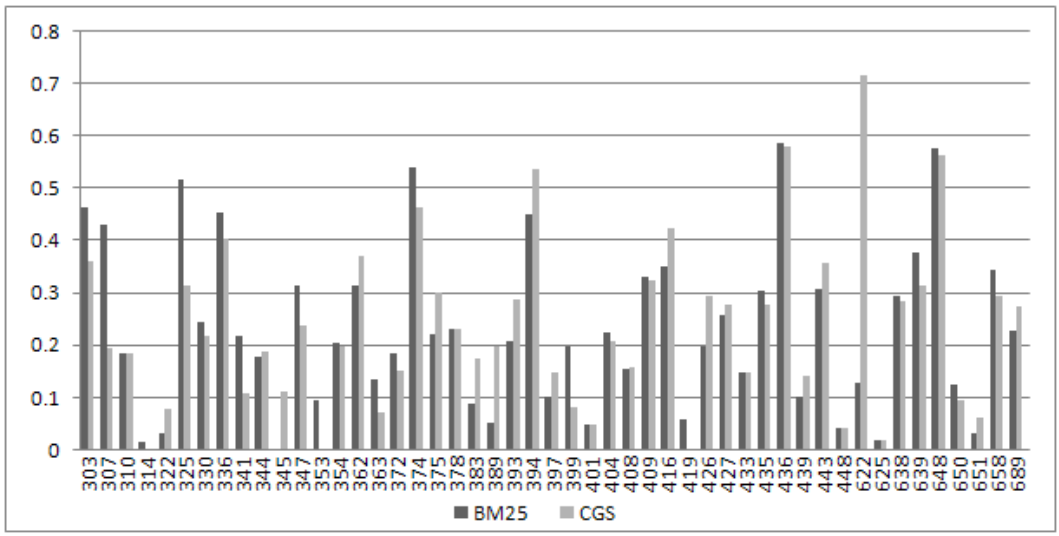
Figure B.2: Query-by-query retrieval performance of CGS on HARD04



(a) Topic number vs. MAP

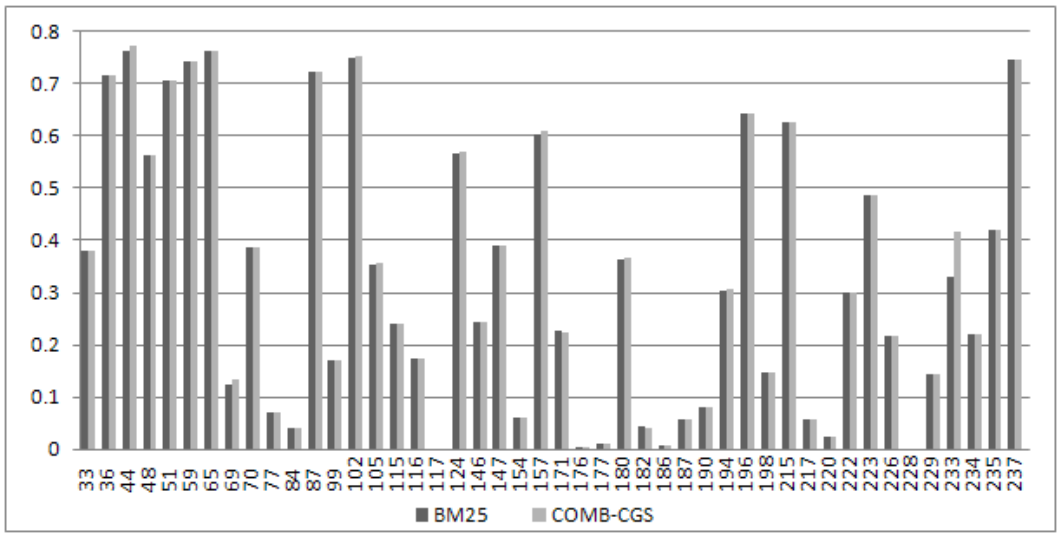


(b) Topic number vs. P10

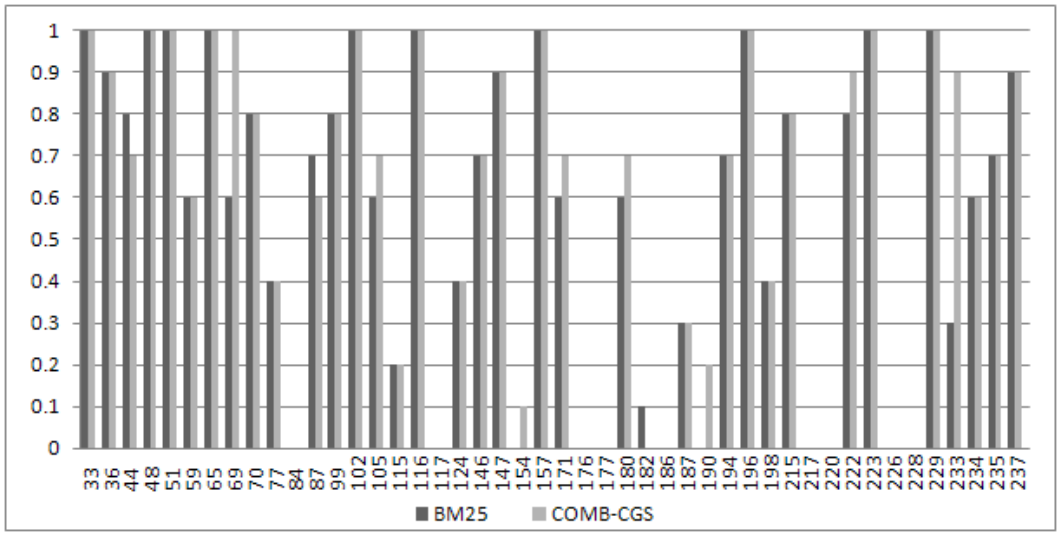


(c) Topic number vs. R-PREC

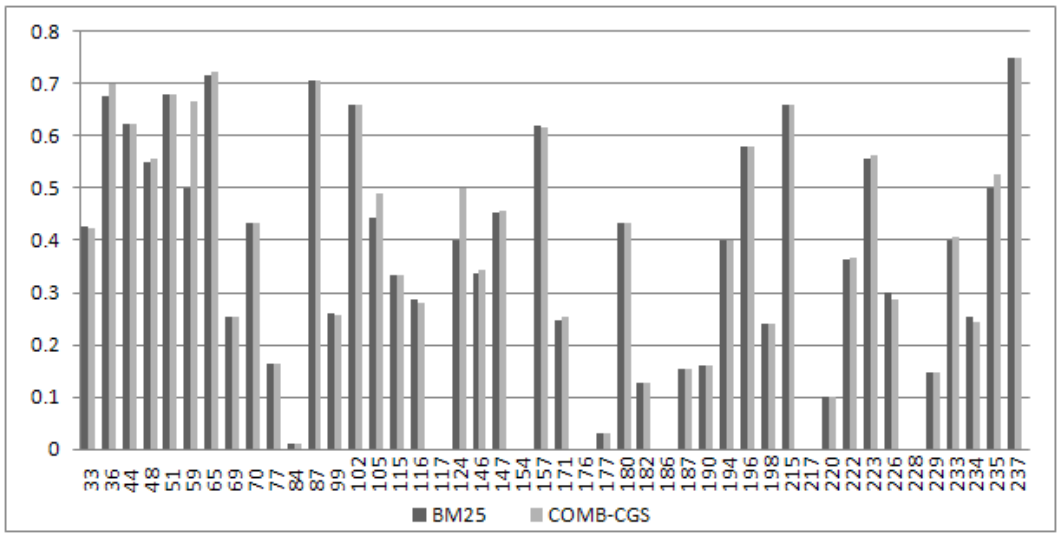
Figure B.3: Query-by-query retrieval performance of CGS on HARD05



(a) Topic number vs. MAP

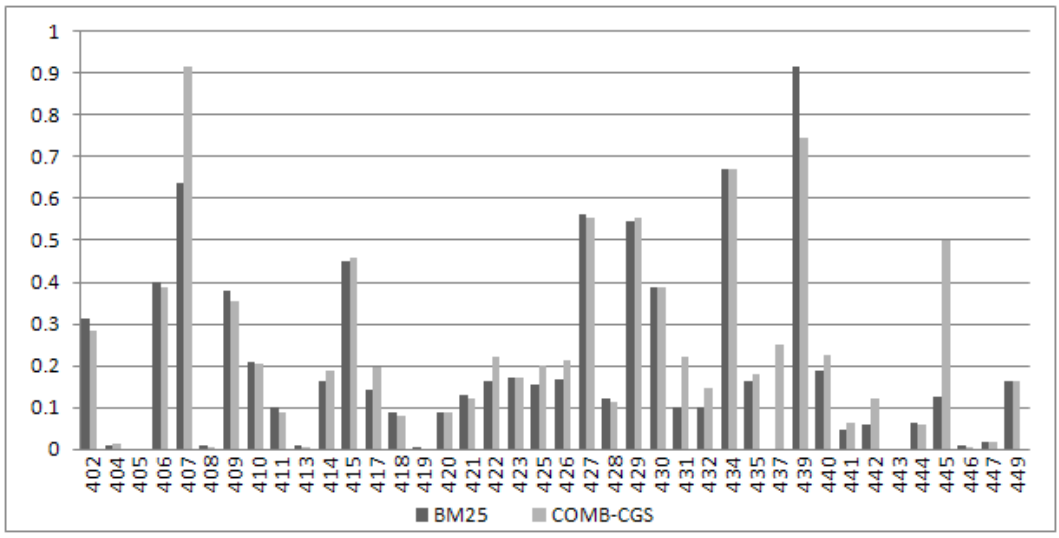


(b) Topic number vs. P10

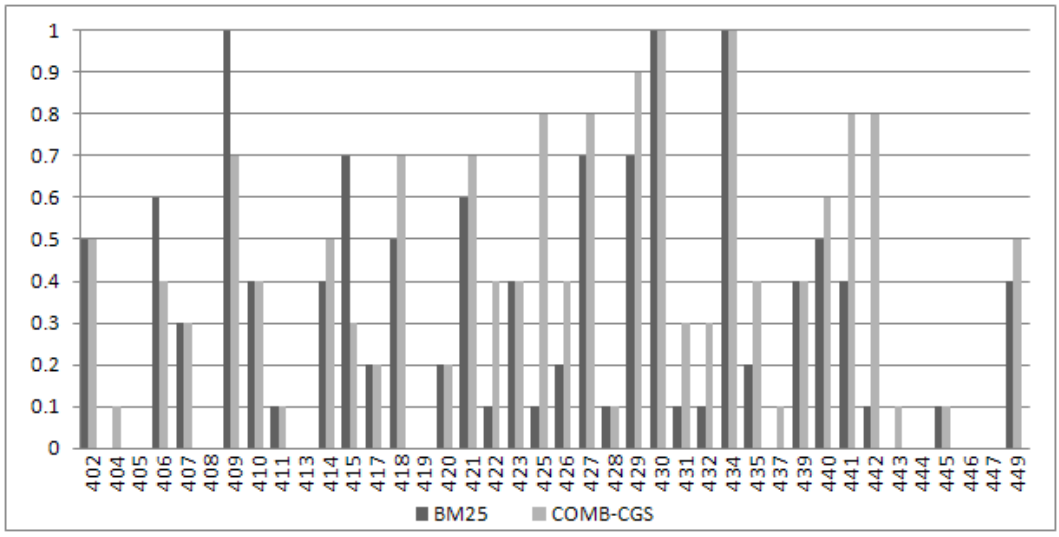


(c) Topic number vs. R-PREC

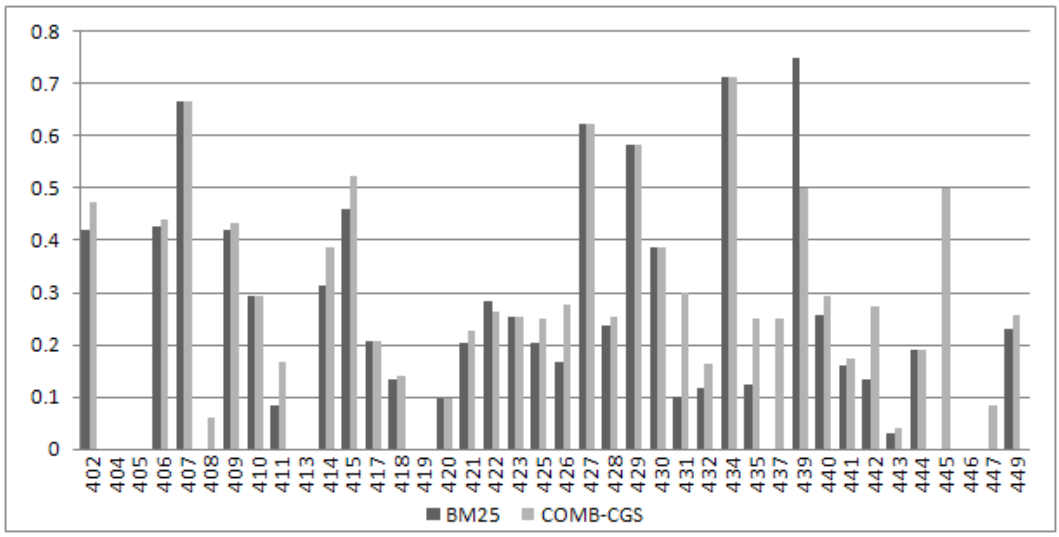
Figure B.4: Query-by-query retrieval performance of COMB-CGS on HARD03



(a) Topic number vs. MAP

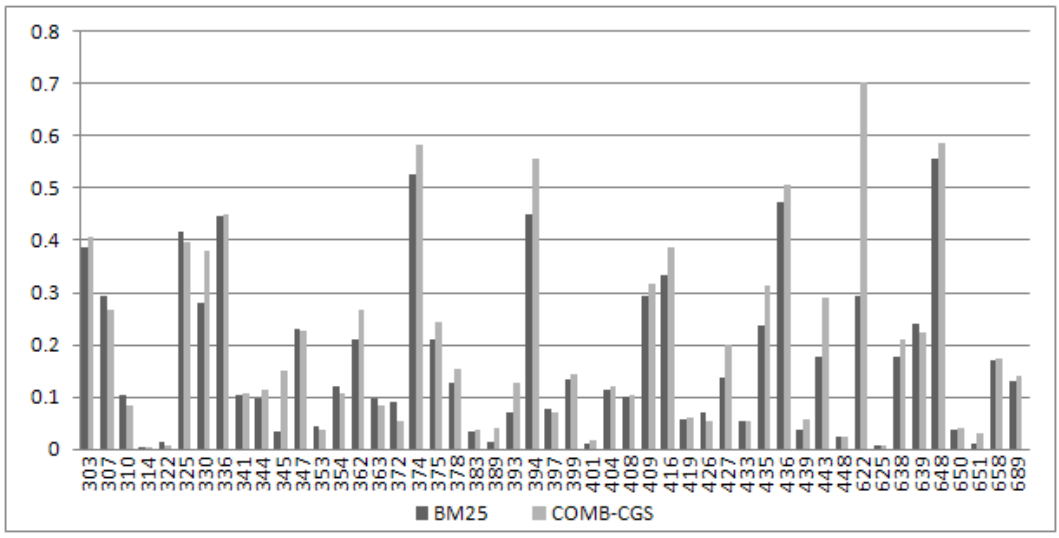


(b) Topic number vs. P10

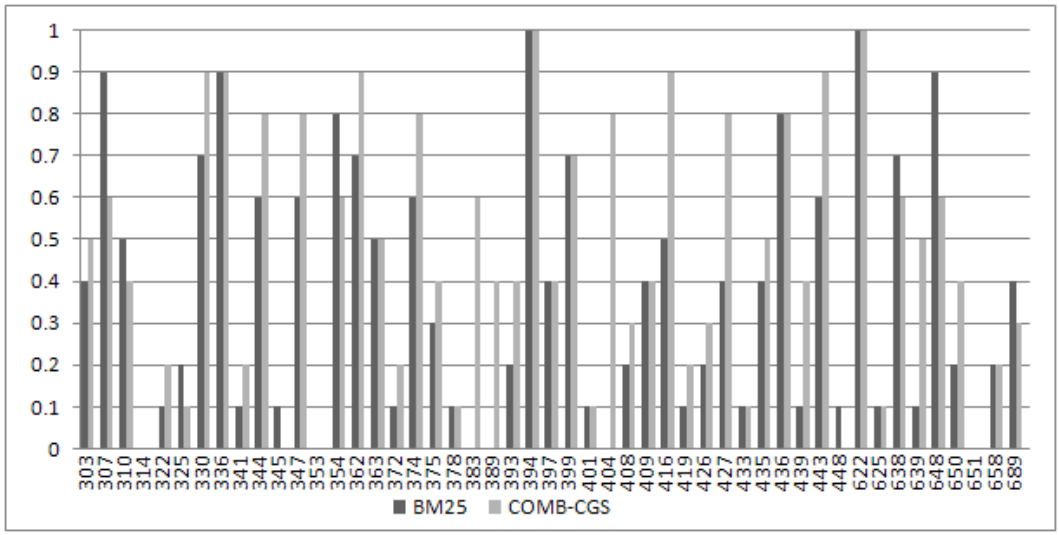


(c) Topic number vs. R-PREC

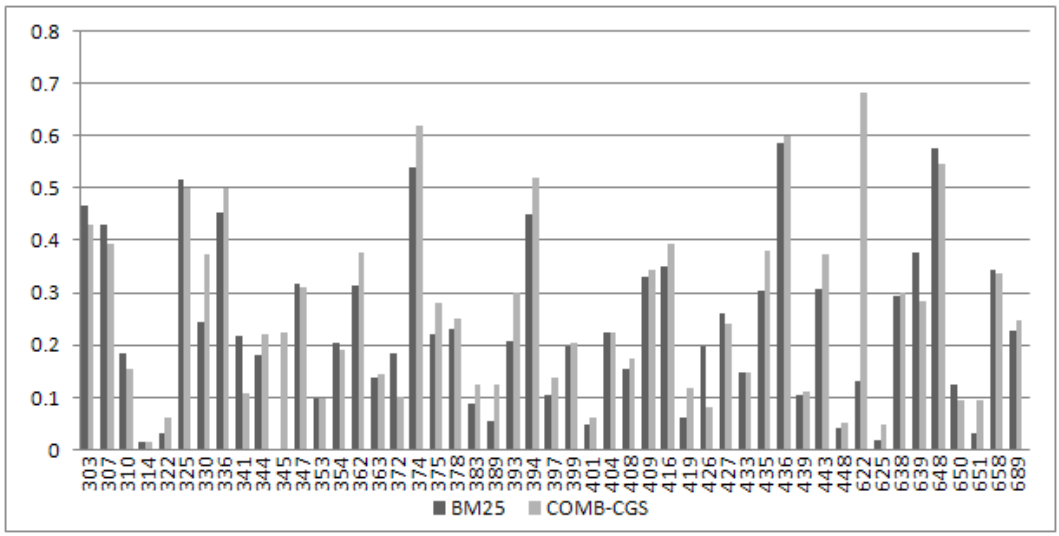
Figure B.5: Query-by-query retrieval performance of COMB-CGS on HARD04



(a) Topic number vs. MAP

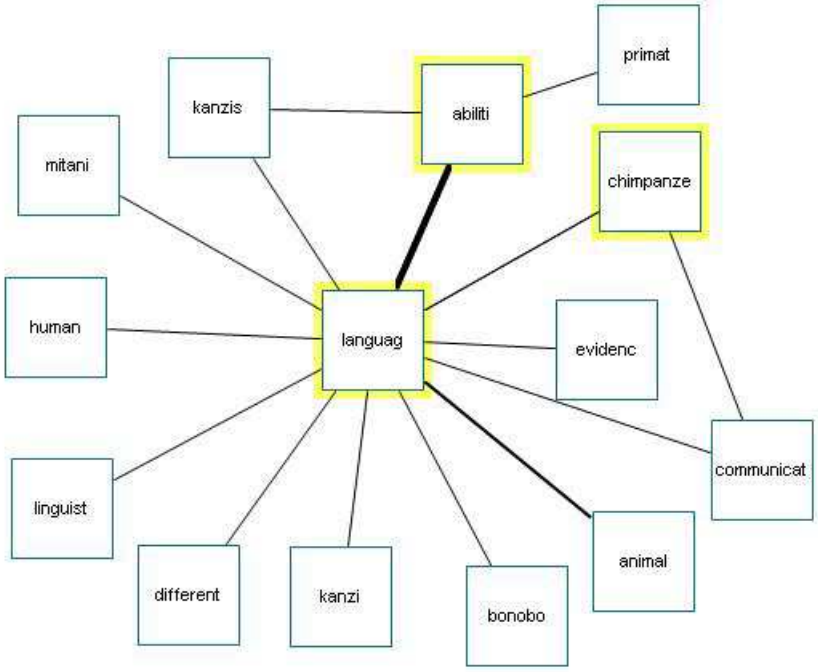


(b) Topic number vs. P10

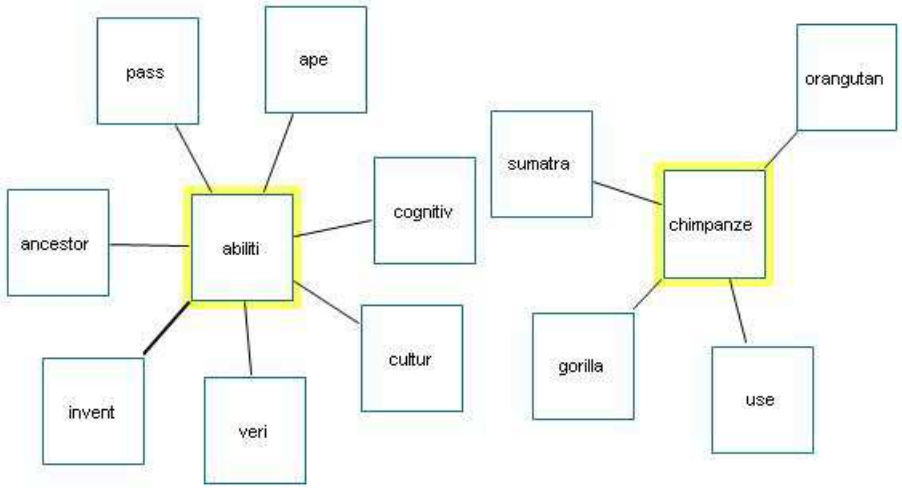


(c) Topic number vs. R-PREC

Figure B.6: Query-by-query retrieval performance of COMB-CGS on HARD05



(a) A relevant document (NYT20030103.0110) ranked 4th by BM25



(b) A non-relevant document (APE20030102.0060) ranked 5th by BM25

Figure B.7: Visual representation of two documents using the Cohesion Graph (F50S1) for the query "Chimpanzee language ability" (HARD-407). The thickness of arcs represents the strength of association between the nodes (i.e. terms). CGS demotes document represented in (b), and promotes document represented in (a)

```
<DOC>
<DOCNO> APW19980601.0821 </DOCNO>
<DOCTYPE> NEWS STORY </DOCTYPE>
<DATE_TIME> 1998-01-06 12:36:00 </DATE_TIME>
<HEADER>
w1066 &Cxlf; wstm-
u s &Cxl3;  &Cxll;  BC-Sports-FieldHockey-Wo     06-01 0178
</HEADER>
<BODY>
<SLUG> BC-Sports-Field Hockey-World Cup,0177 </SLUG>
<HEADLINE>
Netherlands beats Spain 3-2 in overtime for World Cup title
</HEADLINE>
<TEXT>
       UTRECHT, Netherlands (AP) _ Cheered on at home by a stadium
filled with frenzied fans, the Netherlands defeated Spain 3-2 in
sudden-death overtime Monday to win its third men's field hockey
World Cup.
       Olympic champion Netherlands, twice winner of the World Cup in
1975 at Amsterdam and 1990 at Lahore, lifted the title following
Teun de Nooijer's golden goal in the 13th minute of overtime.
       Spain took the lead in the 18th minute through a field goal by
Javier Arnau.
       A penalty corner conversion by Victor Pujol increased Spain's
lead before skipper Stephan Veen pulled one back for the
Netherlands in the 58th minute.
       Penalty corner specialist Bram Lomans scooped the ball into the
cage in the 61st minute to give the Dutch the equalizer and take
the game into extra time.
       De Nooijer slammed a penalty corner rebound after goalkeeper
Ramon Jufresa had blocked the shot from Lomans with two minutes of
extra time remaining.
</TEXT>
(PROFILE
(WS SL:BC-Sports-Field Hockey-World Cup; CT:s;
(REG:MEST;)
(REG:ENGL;)
(REG:AFRI;)
(REG:ASIA;)
(LANG:ENGLISH;))
)
</BODY>
<TRAILER>
AP-NY-06-01-98 1236EDT
</TRAILER>
</DOC>
```

Figure B.8: An example TREC document

```
<topic>

<number>
303
</number>

<title>
Hubble Telescope Achievements
</title>

<description>
Identify positive accomplishments of the Hubble telescope since it
was launched in 1991.
</description>

<topic-narrative>
Documents are relevant that show the Hubble telescope has produced
new data, better quality data than previously available, data that
has increased human knowledge of the universe, or data that has led
to disproving previously existing theories or hypotheses.  Documents
limited to the shortcomings of the telescope would be irrelevant.
Details of repairs or modifications to the telescope without
reference to positive achievements would not be relevant.
</topic-narrative>

</topic>
```

Figure B.9: An example TREC topic

```
num_q             all 49
num_ret           all 49000
num_rel           all 6466
num_rel_ret       all 3904
map               all 0.1697
gm_ap             all 0.0981
R-prec            all 0.2307
bpref             all 0.2189
recip_rank        all 0.5634
ircl_prn.0.00     all 0.6166
ircl_prn.0.10     all 0.3679
ircl_prn.0.20     all 0.2985
ircl_prn.0.30     all 0.2368
ircl_prn.0.40     all 0.1936
ircl_prn.0.50     all 0.1451
ircl_prn.0.60     all 0.1075
ircl_prn.0.70     all 0.0832
ircl_prn.0.80     all 0.0561
ircl_prn.0.90     all 0.0274
ircl_prn.1.00     all 0.0033
P5                all 0.4122
P10               all 0.3694
P15               all 0.3592
P20               all 0.3469
P30               all 0.3293
P100              all 0.2384
P200              all 0.1858
P500              all 0.1236
P1000             all 0.0797
```

Figure B.10: A sample trec-eval output

```
HARD-033->Animal Protection
HARD-036->International Year of Older Persons (IYOP)
HARD-044->Amusement Park Safety
HARD-048->Y2K crisis
HARD-051->Hate Crimes Prevention
HARD-059->Alexandria's New Library
HARD-065->Mad Cow disease
HARD-069->Environmental protection
HARD-070->Red Cross activities
HARD-077->Insect-borne illnesses
HARD-084->Recent Earthquakes
HARD-087->Egyptian cotton
HARD-099->Globalization and democracy
HARD-102->Microsoft monopoly
HARD-105->Healthy tea
HARD-115->Virtual defense
HARD-116->Genetic Modification technology
HARD-117->US Civil Unrest
HARD-124->Cell phone health hazard
HARD-146->NATO/UN Tension over Balkans Crisis
HARD-147->Regional Economic Integration
HARD-154->Instant messaging
HARD-157->Public Transit Funding
HARD-171->Global population
HARD-176->Geo-Politics of War
HARD-177->Rewriting Indian history
HARD-180->Euro Introduced
HARD-182->School development
HARD-186->Restricting the Internet
HARD-187->National Leadership Transitions
HARD-190->Climate Change
HARD-194->Suburban Sprawl
HARD-196->IPO Activity
HARD-198->Wartime Propaganda
HARD-215->Laws about hijackers
HARD-217->Iraq disarmament
HARD-220->Future of Mid-East Peace
HARD-222->Corporate Mergers
HARD-223->Sports Scandals
HARD-226->Efficiency of computer operating systems
HARD-228->Child's play
HARD-229->The history of nanotechnology
HARD-233->China Human rights
HARD-234->Global Wage Remittance
HARD-235->Product Customization
```

Figure B.11: HARD03 queries

HARD-402->Identity Theft
HARD-404->Marathon Training
HARD-405->Female competitive fighters
HARD-406->The Diamond Industry
HARD-407->Chimpanzee Language Ability
HARD-408->College Campus Racism
HARD-409->AIDS in Africa
HARD-410->Low-Carb Mania
HARD-411->Natural Disasters and Global Warming
HARD-413->The Future of Corn
HARD-414->Human Evolution
HARD-415->Life on Mars
HARD-417->Diabetes Research
HARD-418->Immigration Post 9-11
HARD-419->Do It Yourself Computer Building
HARD-420->Internet Security through Quantum Computing
HARD-421->Software Patents
HARD-422->Video game crash
HARD-423->United Nations Development Programme's Millennium Declaration
HARD-425->Cancun World Trade Talks
HARD-426->Chinese capitalism
HARD-427->Brazilian Landless Workers Movement
HARD-428->International organ traffickers
HARD-429->Biodynamic and Organic farming
HARD-430->Chlorofluorocarbon Ban
HARD-431->Britney Spears
HARD-432->Farm Subsidies
HARD-434->The Zapatista movement
HARD-435->LeBron James
HARD-437->Role Playing Games
HARD-439->Giant Squid
HARD-440->Supreme Court Rulings
HARD-441->European Union
HARD-442->Interest Rates
HARD-443->Hand-Held Electronics
HARD-444->European Elections
HARD-445->Bird Migration Deaths
HARD-446->Grand Canyon Environment
HARD-447->VX nerve gas disposal
HARD-449->Multiple Pregnancies

Figure B.12: HARD04 queries

HARD-303->Hubble Telescope Achievements
HARD-307->New Hydroelectric Projects
HARD-310->Radio Waves and Brain Cancer
HARD-314->Marine Vegetation
HARD-322->International Art Crime
HARD-325->Cult Lifestyles
HARD-330->Iran-Iraq Cooperation
HARD-336->Black Bear Attacks
HARD-341->Airport Security
HARD-344->Abuses of E-Mail
HARD-345->Overseas Tobacco Sales
HARD-347->Wildlife Extinction
HARD-353->Antarctica exploration
HARD-354->journalist risks
HARD-362->human smuggling
HARD-363->transportation tunnel disasters
HARD-372->Native American casino
HARD-374->Nobel prize winners
HARD-375->hydrogen energy
HARD-378->euro opposition
HARD-383->mental illness drugs
HARD-389->illegal technology transfer
HARD-393->mercy killing
HARD-394->home schooling
HARD-397->automobile recalls
HARD-399->oceanographic vessels
HARD-401->foreign minorities, Germany
HARD-404->Ireland, peace talks
HARD-408->tropical storms
HARD-409->legal, Pan Am, 103
HARD-416->Three Gorges Project
HARD-419->recycle, automobile tires
HARD-426->law enforcement, dogs
HARD-427->UV damage, eyes
HARD-433->Greek, philosophy, stoicism
HARD-435->curbing population growth
HARD-436->railway accidents
HARD-439->inventions, scientific discoveries
HARD-443->U.S., investment, Africa
HARD-448->ship losses
HARD-622->price fixing
HARD-625->arrests bombing WTC
HARD-638->wrongful convictions
HARD-639->consumer on-line shopping
HARD-648->family leave law
HARD-650->tax evasion indicted
HARD-651->U.S. ethnic population
HARD-658->teenage pregnancy
HARD-689->family-planning aid

Figure B.13: HARD05 queries