Concept-Based Information Retrieval using Explicit Semantic Analysis

Ofer Egozi

Technion - Computer Science Department - M.Sc. Thesis MSC-2010-06 - 2010

Concept-Based Information Retrieval using Explicit Semantic Analysis

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Ofer Egozi

Submitted to the Senate of
the Technion — Israel Institute of Technology
Heshvan 5770 Haifa November 2009

The research thesis was done under the supervision of Prof. Shaul Markovitch in the Computer Science Department.

Contents

List of Figures

Abstract

Information Retrieval systems traditionally rely on textual keywords to index and retrieve documents. Keyword-based retrieval may return inaccurate and incomplete results when different keywords are used to describe the same human concept in the documents and in the queries. Furthermore, the relationship between those keywords may be semantic rather than syntactic, and capturing it thus requires access to comprehensive human world knowledge. Concept-based retrieval methods have attempted to tackle these difficulties by using manually-built thesauri, by relying on term co-occurrence data, or by extracting latent word relationships and concepts from a corpus. In this paper we introduce a new concept-based retrieval method that is based on Explicit Semantic Analysis (ESA). ESA is a recently proposed representation method that can augment the keyword-based representation with concept-based features, automatically extracted from massive human knowledge resources such as Wikipedia. We have found that high-quality feature selection is required to make the retrieval more focused. However, due to the lack of labeled data, traditional statistical filtering methods cannot be used. We introduce several selection methods that use self-generated labeled training data. The resulting system is evaluated on TREC data, showing superior performance over previous state-of-the-art results.

Abbreviations and Notations

IR	—	Information Retrieval
ESA	—	Explicit Semantic Analysis
BOW	—	Bag of Words
TC	—	Text Categorization
TF	—	Term Frequency
MAP	—	Mean Average Precision
FS	—	Feature Selection
PRF	—	Pseudo-Relevance Feedback
IG	—	Information Gain
IIG	—	Incremental Information Gain
RV	—	Rocchio's Vector

Chapter 1

Introduction

Information Retrieval (IR) systems are concerned with providing the most relevant documents to a user's query. With early IR systems used mainly by retrieval experts, initial IR methodology was based on keywords manually assigned to documents, and on complicated Boolean queries. As automatic indexing and natural language queries gained popularity in the 1970s, IR systems became increasingly more accessible to non-expert users. Documents were indexed by automatically considering all terms in them as independent keywords, in what is known as the Bag-of-Words (BOW) representation, and query formatting was simplified to a short natural language formulation. However, even as the keywords became "noisier," the basic methodology for indexing them remained unchanged. Thus, these non-expert users were increasingly faced with what was described as "the vocabulary problem" [15]. The keywords chosen by the users were often different from those used by the authors of the relevant documents, lowering the systems' *recall* rates. In other cases, the contextual differences between ambiguous keywords were overlooked by the BOW approach, reducing the *precision* of the results. These two problems are commonly referred to as *synonymy* and *polysemy*, respectively.

IR researchers attempted to resolve the synonymy problem by expanding the original query with synonyms of query keywords [52]. However, the relationship between the keywords chosen by the users and those used by the authors often extends beyond simple synonymy. Consider the short query "Estonia economy," an actual query (#434) in the TREC-8 Adhoc test collection [55]. A relevant document may discuss announcements by the

ministry of trade in Tallinn (the Estonian capital), with no mention of any direct synonym of any of the query keywords.

To handle such problems, new query expansion methods that rely on corpus-based evidence were suggested. For example, [57] suggested identifying terms that co-occur with query keywords in the top-ranked documents for the query, to be used as expansion terms that are more broadly related to the query (such as "trade" and "Tallinn," in this example). Such approaches showed significant improvement, but require manual tuning in order not to adversely affect performance: too few expansion terms may have no impact, and too many will cause a *query drift* [37].

For tackling polysemy, the main proposed method was to apply automatic word sense disambiguation algorithms to the documents and query. Disambiguation methods use resource such as the Wordnet thesaurus [51] or co-occurrence data [45] to find the possible senses of a word and map word occurrences to the correct sense. These disambiguated senses are then used in indexing and in query processing, so that only documents that match the correct sense are retrieved. The inaccuracy of automatic disambiguation is the main obstacle in achieving significant improvement using these methods, as incorrect disambiguation is likely to harm performance rather than merely not improve it.

Concept-based information retrieval is an alternative IR approach that aims to tackle these problems differently. Concept-based IR represents both documents and queries using semantic *concepts*, instead of (or in addition to) keywords, and performs retrieval in that concept space. This approach holds the promise that representing documents and queries using high-level concepts will result in a retrieval model that is less dependent on the specific terms used. Such a model could yield matches even when the same notion is described by different terms in the query and target documents, thus alleviating the synonymy problem and increasing recall. Similarly, if the correct concepts are chosen for ambiguous words appearing in the query and in the documents, non-relevant documents that were retrieved with the BOW approach could be eliminated from the results, thus alleviating the polysemy problem and increasing precision.

Existing concept-based methods can be characterized by the following three parameters:

1. Concept representation – the "language" the concepts are based on.

Explicit real-life concepts [51, 19] better lend themselves to human interpretation and reasoning than do implicit or latent concepts [10, 25, 59]. However, sufficient coverage and granularity are major challenges.

2. Mapping method – the mechanism that maps natural language texts to these concepts. The most accurate mechanism would likely be manual, building a hand-crafted ontology of concepts with a list of words to be assigned to each [35], but such an approach involves huge effort and complexity. The mapping can also be automatic, using machine learning [19], though that would usually imply less accurate mapping.

3. Use in IR – the stages in which the concepts are used. Concepts would be best used throughout the entire process, in both indexing and retrieval stages [20]. A simpler but less accurate solution would apply concept analysis in one stage only, as in concept-based query expansion [40].

Of all the approaches suggested so far for concept-based IR, none offers an optimal combination of the above choices. An ideal approach would use explicit semantic concept representation, with no limits on domain coverage or conceptual granularity, would support a fully-automatic mechanism for mapping texts onto those concepts, would be computationally feasible even for very large corpora, and would integrate concept-based processing in both indexing and retrieval stages.

In this paper we propose a novel concept-based IR approach that meets all of the above requirements, using Explicit Semantic Analysis (ESA). The concepts used are taken from a very comprehensive, human-defined ontology of explicit concepts. Text analysis methods are used to automatically and efficiently extract these concepts and represent any document or query text using them. Finally, the proposed system builds upon existing IR methodology and integrates concepts into both document indexing and retrieval, using standard data structures and ranking methods.

We show that a naive implementation of IR using these concepts is insufficient, due to the concepts' inherent noisy nature. We address these difficulties by embedding feature selection methods into the retrieval process, and we further improve the results by combining the concept-based results with those of keyword-based retrieval. We evaluate the proposed system on TREC datasets to show significant improvement in performance

compared both with our own baseline and with published results of other state-of-the-art systems.

Our main contributions in this work are threefold: a framework for using the ESA representation method in information retrieval, a method for integrating feature selection into the concept-based IR task, and three selection methods that are based on common AI methods and shown to be beneficial for the task at hand.

The remainder of this paper is organized as follows. Section 2 provides background on ESA. Sections 3 to 5 describe the proposed concept-based algorithms and empirical evaluation results. Section 6 surveys related work on concept-based IR, and Section 7 concludes the paper.

Chapter 2

Explicit Semantic Analysis (ESA)

Explicit Semantic Analysis, or ESA [17], is a recently proposed method for semantic representation of general-domain natural language texts. ESA represents meaning in a high-dimensional space of concepts, automatically derived from large-scale human-built repositories such as Wikipedia[1]. Since it was first proposed, ESA has been successfully applied to text categorization [17, 21, 7], semantic relatedness calculation [18, 22], cross-language information retrieval [39, 49], and concept-based information retrieval [12].

In Wikipedia-based ESA, the semantics of a given word are described by a vector storing the word's association strengths to Wikipedia-derived concepts. A concept is generated from a single Wikipedia article, and is represented as a vector of words that occur in this article weighted by their tf.idf score. Once these concept vectors are generated, an inverted index is created to map back from each word to the concepts it is associated with. Thus, each word appearing in the Wikipedia corpus can be seen as triggering each of the concepts it points to in the inverted index, with the attached weight representing the degree of association between that word and the concept. The process is illustrated in Figure 2.1.

With this resource in hand, any input word to a text processing task can now be semantically represented as a sparse vector in the high-dimensional space of Wikipedia concepts. Larger text fragments are represented as a

[1] http://www.wikipedia.org

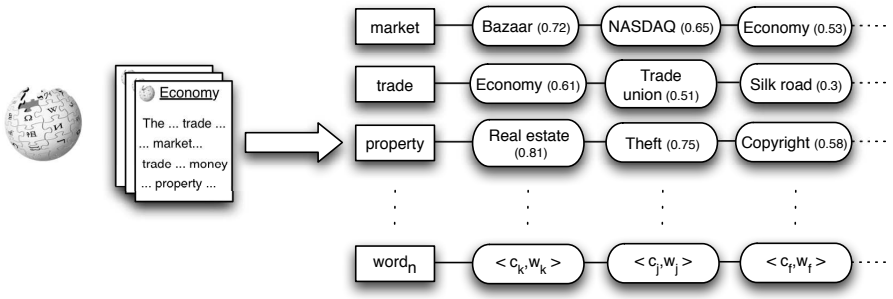

Figure 2.1: Generation of an ESA model from Wikipedia articles. The articles and words in them are processed to build a weighted inverted index, representing each word as a vector in the space of all Wikipedia concepts (articles).

concept vector that is a combination of the separate vectors of its individual terms, and ESA operations can then be carried out by manipulating these vectors. For example, computing semantic relatedness between two texts can be reduced to generating the ESA concept vectors for each of them, and then calculating their cosine similarity.

To illustrate the nature of ESA concepts, we show the top concepts generated by our ESA implementation for two short news clip fragment:

- **Text:** *"A group of European-led astronomers has made a photograph of what appears to be a planet orbiting another star. If so, it would be the first confirmed picture of a world beyond our solar system."*

  Top generated concepts: (1) Planet; (2) Planetary orbit; (3) Solar system; (4) Extrasolar planet; (5) Jupiter; (6) Astronomy; (7) Definition of planet; (8) Pluto; (9) Minor planet; (10) PSR 1257+12

  All concepts are highly relevant and describe or relate to the subject of the text, with the fourth concept (Extrasolar planet) being the exact topic, despite the fact that these words were not explicitly mentioned in the text. PSR 1257+12 is the name of a pulsar around which the first extrasolar planets were discovered orbiting.

- **Text:** *"New Jaguar model unveiled by firm"*

Top generated concepts: (1) Jaguar XJ; (2) Jaguar (car); (3) Ford Motor Company; (4) Jaguar XK; (5) Land Rover Range Rover; (6) Jaguar S-Type; (7) Jaguar X-Type; (8) Nissan Micra; (9) V8 engine; (10) Jaguar E-type

This example demonstrates the disambiguation power of ESA, as the top concepts all refer to Jaguar the car maker rather than to the namesake animal or American football team. Despite the text containing no explicit car-related terms, words such as "model" and "unveil" were more related to the industry meaning and helped trigger the correct concepts. The concepts generated also hint at rich *world knowledge*, such as the business relations to Ford Motor Company and Land Rover Range Rover and the use of a V8 Engine on Jaguar models. The Nissan Micra concept was triggered by a Micra variant that was inspired by a Jaguar model.

The use of a knowledge repository as large and diverse as Wikipedia creates a powerful concept ontology, well suited for concept-based IR. First, Wikipedia's broad coverage of a huge range of topics, coupled with automatic ontology-building, yields a highly fine-grained ontology. Second, the language coverage of the inverted index, mapping from a massive aggregation of natural language terms (the entire Wikipedia corpus) to the concepts in which they occur, produces a powerful classifier to automatically map any given text fragment to this concept ontology. Finally, the use of a semantics-based ontology such as Wikipedia's, or in another implementation the Open Directory Project [16], generates meaningful and human-readable concepts that can provide additional reasoning for the researcher and for system users.

Chapter 3

ESA-Based Retrieval

Given the described advantages of ESA as a semantic representation and its demonstrated success in other text analysis tasks, it appears well suited for building a successful concept-based IR model. In this section we introduce our first algorithm for concept-based IR using ESA representation. The algorithm maps documents and queries to the Wikipedia-ESA concept space, and performs indexing and retrieval in that space. We then evaluate the algorithm's performance on TREC datasets. We show that combining concept-based relevancy of documents with that of passages in these documents, performs best for ESA-based retrieval. We also find that the quality of generated concepts is lower than expected, and analyze the potential causes and remedies to be applied in the next section.

3.1 ESA Concept-Based Indexing

We use ESA to map each document in the corpus to a weighted vector of concepts. Like BOW vectors, concept-based vectors are also sparse, with concept weights being zero for most of the Wikipedia concepts. Nevertheless, given that each word in the document to be indexed may still be related to a large number of concepts, and that a document containing a collection of words is likely to be related to an even larger number, indexing the entire list of related concepts for every document is not feasible. We therefore use only the concepts with the highest weights (association scores). In a sorted representation of the vector, this subset of concepts is simply its prefix.

Long documents are more difficult to map in full into the ESA concept space. A small part of a long document might be relevant to the current query, but the semantics of this part may be underrepresented in the concepts vector for the full document. A similar problem exists also in BOW approaches, where the term frequency (TF) measure must be normalized [48] to account for documents of different lengths. However, for concept-based retrieval the challenge is even greater: because of the averaging effect of the representation of longer text fragments and the practical need to use only a small subset of the representation concepts, the concepts of the relevant section might be pruned out of the indexed vector.

Previous research using BOW representation has shown that breaking long documents into shorter *passages* can improve document retrieval [5], with the ranking of passages viewed as evidence to the relevance of their source documents. Furthermore, it has been shown that *fixed-length* passages yield better results than passages based on syntactic or semantic segmentation [5, 28]. We therefore suggest a similar approach, breaking documents into length-based overlapping passages and representing each passage separately by its own generated set of concepts. We expect such an approach to achieve better results, in particular with long documents that cover several themes.

Note that while [17] also split documents into sentence and paragraph contexts in applying ESA to text categorization, they eventually combined the concepts of these sub-contexts into a single unified representation. In our approach, each passage is indexed and may be retrieved as a stand-alone unit of information. Thus, a passage is ranked separately as an independent indicator of its original document's relevance.

We now have, for any document to be indexed, a set of passages and a concept vector representation for each. We index these concepts in a standard IR inverted index, using the concepts' unique identifiers as tokens. The score associated with each concept in the vector is used as the token weight, equivalent to term frequency in standard text indexing. The pseudocode for the above indexing algorithm is described in Figure 3.1.

```
#Index corpus 𝒟 using ESA concepts; trim ESA vector to the s
# first concepts, segment documents to passages of length l
Procedure ESA-Indexing(𝒟, s, l)
    Foreach d ∈ 𝒟
        F⃗_d ← ESA(d, s)
        Foreach ⟨c_i, w_i⟩ ∈ F⃗_d
            add ⟨d, w_i⟩ to InvIndex[c_i]
        𝒫_d ← Divide-into-Passages(d, l)
        Foreach p ∈ 𝒫_d
            F⃗_p ← ESA(p, s)
            Foreach ⟨c_i, w_i⟩ ∈ F⃗_p
                add ⟨p, w_i⟩ to InvIndex[c_i]
```

Figure 3.1: ESA-based indexing in an inverted index

3.2 ESA-Based Retrieval Algorithm

Upon receiving a query, our algorithm first converts it to an ESA concept vector. The representation method is identical to the one by which documents and passages are represented at index time. Having indexed full documents and passages, we now have to choose how these two types of evidence are to be combined for ranking. Following [5], we retrieve both sets of results and sum each document's full score with the score of the best performing passage in it[1]. The documents are then sorted by this combined score and the top scoring documents are output, as described[2] in Figure 3.2.

The retrieval algorithm has a single parameter s controlling the cutoff (as described in the previous section) of the query concept vector. The value for s may be chosen to be the same as that in the indexing process, but not necessarily. Indexing the entire corpus with large cutoff values would incur significant storage and computation costs, and is therefore not

[1] We also experimented with assigning different weights to these two summed scores but found no improvement in doing so

[2] In practice, the retrieval process is optimized to not iterate on all indexed documents; hence this combination is performed only for the top ranking documents (the top 1000 in our case), but the principle is similar.

#Retrieve ESA concept-based results for query $\vec{q}$, cutoff
 # concept vector at s
Procedure ESA-RETRIEVAL($\vec{q}, s$)
    $\vec{F_q} \leftarrow \text{ESA}(\vec{q}, s)$
    Return DOCSPASS-RETRIEVE($\vec{F_q}$)

#Retrieve results for query $\vec{q}$ from the combined index.
#INVINDEX-SCORE() stands for the standard inverted index
#function that scores a document's match to a query
Procedure DOCSPASS-RETRIEVE($\vec{q}$)
    Foreach $d \in \mathcal{D}$
        $W_d \leftarrow$ INVINDEX-SCORE($\vec{q}, d$)
        Foreach $p \in$ PASSAGES(d)
            $W_p \leftarrow$ INVINDEX-SCORE($\vec{q}, p$)
        $W'_d \leftarrow W_d + \max W_p$
    Return ranked list according to W'_d

Figure 3.2: ESA-based retrieval

feasible. The query representation, on the other hand, being derived from a much shorter text fragment and incurring no such costs, could benefit from a finer representation, using a higher value for s.

3.3 Empirical Evaluation

In order to evaluate the usefulness of ESA concept-based retrieval, we carried out a set of experiments.

3.3.1 Implementation

We used Xapian[3], an open source probabilistic IR library, as the basis for our experimental platform. Document keywords and concepts were indexed in a Xapian inverted index. In addition, Xapian's implementation of the popular Okapi BM25 ranking formula [42] served as a BOW baseline. Most of the experiments used the TREC-8 Adhoc [55] and the TREC Robust 2004

[3] http://xapian.org/

[53] datasets. The TREC-8 dataset consists of 528,000 documents (mostly newswire) and 50 topics (information-need descriptions, to be transformed into queries), and the Robust-04 dataset uses the same document collection with a different set of 49 topics. We used only the short ("title") queries in TREC topics, since these short (1-3 words) queries better represent common real-life queries, in particular on the Web [2], and since short texts stand best to benefit from conceptual representation [38]. We use the Mean Average Precision (MAP) evaluation measure, commonly used by TREC participants [55], which combines precision and recall while assigning higher importance to the higher-ranking relevant documents.

Documents and passages were stemmed, stopped and indexed by their BOW representation, to serve as the keyword baseline index. Then, ESA-based representations were created and indexed separately as described in Figure 3.1. Passages were set to be fixed-size overlapping segments, shown to be most effective by [28], with passage size set to 50 words. We also tried to use longer passages (200 words) but this proved to be less effective.

The ESA implementation used in our experiments is as described in [17], with ESA vector cutoff in the indexing stage (s in Figures 3.1 and 3.2) set to 50 concepts for practical reasons (index size)[4].

3.3.2 Results

Figure 3.3 shows the performance (MAP) of our ESA-based retrieval algorithm for various parameter values. To assess the impact of the concept vector truncation, we measured performance for varying values of s (the ESA vector cutoff level) in the *query* vector. In addition, to validate the added value of combining documents and passages scores, we compared performance of the combined score to that of documents and passages alone.

As Figure 3.3 clearly shows, passage context outperforms document context significantly, but the best results are achieved when both are combined, an outcome that is consistent with previous IR findings for BOW representations [8]. We will be using the combined documents+passages scoring from here onwards.

Results for increasing values of s indicate that merely adding lower-

[4]We have also experimented with indexing the 100 strongest concepts instead of the 50 strongest, and found no significant impact on the performance.

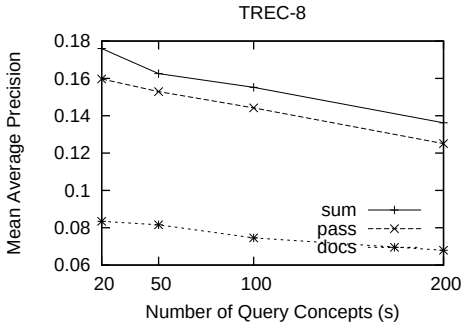

Figure 3.3: ESA-based retrieval performance as a function of ESA cutoff and ranking contexts

ranking concepts in the ESA vector does not improve retrieval. Not only does the precision-oriented MAP score decrease as concepts are added, but the absolute recall (measured in the top 1000 retrieved documents) decreases as well. This finding suggests that some of the generated concepts may be detrimental, and that successful application of ESA to IR may require further selection of the concepts initially generated for the query. We will revisit this hypothesis later on.

However, even when choosing the best performing parameter values, ESA-based retrieval (MAP of 0.1760) is significantly inferior to that of our BOW baseline (MAP of 0.2481). Considering the superior results obtained when ESA-based representation was applied to previous text analysis applications ([17, 18]), this result is quite surprising. In the following subsection, we conduct a qualitative analysis of specific retrieval cases in order to better understand the causes of this inferior retrieval and to suggest ways to remedy them.

3.3.3 Qualitative Analysis

The results shows that ESA-based retrieval can indeed, as expected, identify relevant documents even when these do not include query terms or their simple synonyms. Let us consider TREC query 411 ("salvaging shipwreck treasure"). The following short relevant document was retrieved by the ESA-based method but not by the BOW baseline:

> *"ANCIENT ARTIFACTS FOUND. Divers have recovered artifacts lying underwater for more than 2,000 years in the wreck of a Roman ship that sank in the Gulf of Baratti, 12 miles off the island of Elba, newspapers reported Saturday."*

The top 10 concepts generated for this document were:

Scuba diving
Wreck diving
RMS Titanic
USS Hoel (DD-533)
Shipwreck
Underwater archaeology
USS Maine (ACR-1)
Maritime archaeology
Tomb Raider II
USS Meade (DD-602)

whereas the query's top 10 concepts were:

Shipwreck
Treasure
Maritime archaeology
Marine salvage
History of the British Virgin Islands
Wrecking (shipwreck)
Key West, Florida
Flotsam and jetsam
Wreck diving
Spanish treasure fleet

With 3 matches in the top-10 concepts (and more in lower positions), the ESA-based method was capable of retrieving this relevant document as its third ranked result, despite the fact that not one of the query terms appears in the document's text.

Let us now examine a contrary example, where concept-based retrieval returned a *non-relevant* document, one that was not returned by the BOW baseline. We revisit query 434 ("Estonia economy"), for which the following short document was retrieved using the concept-based method:

"Olympic News In Brief: Cycling win for Estonia. Erika Salumae won Estonia's first Olympic gold when retaining the women's cycling individual sprint title she won four years ago in Seoul as a Soviet athlete."

Although this document is Estonia-related, it concerns not economy but sports. The document's top 10 concepts were:

Estonia at the 2000 Summer Olympics
Estonia at the 2004 Summer Olympics
2006 Commonwealth Games
Estonia at the 2006 Winter Olympics
1992 Summer Olympics
Athletics at the 2004 Summer Olympics - Women's Marathon
2000 Summer Olympics
2006 Winter Olympics
Cross-country skiing at the 2006 Winter Olympics
New Zealand at the 2006 Winter Olympics

The concepts seem quite relevant, discussing Estonia and various Olympics-related themes. Now let us examine the query's top 10 concepts:

Estonia
Economy of Estonia
Estonia at the 2000 Summer Olympics
Estonia at the 2004 Summer Olympics
Estonia national football team
Estonia at the 2006 Winter Olympics
Baltic Sea
Eurozone
Tiit Vähi
Military of Estonia

Technically, this document was correctly retrieved by the system, with three of the top concepts shared between query and document. But why were these sports-related concepts generated for this query, despite the query's bearing no relation whatsoever to sports?

The Wikipedia articles from which these sports-related concepts were derived contain no mention of the word "economy," but do contain many

instances of the word "Estonia." Thus, the tf.idf score used to compute the weight for the word "Estonia" in these sports-related concepts was very high. Hence, even when the query contains other words (such as "economy") for which the weight of these sports-related concepts is very low, the ESA vector for the entire query still assigns them a high weight. As a result, Estonian sports-related documents are ranked too high and are incorrectly retrieved by the system, degrading overall performance. The query concept vector does include concepts related to Estonia's economy, such as Economy of Estonia, Tiit Vähi (Estonia's prime minister during the country's major economic transformation period) and Eurozone, but these are not effective in removing the non-relevant sports results. In this respect, the effect is similar to that of query drift [37] that is caused by excessive text-based query expansion.

Our observation, then, is that since the ESA classifier is created from a noisy unstructured information source, and one that is different from the target corpus, the initial concept vector might carry noise and ambiguities. To counter such problems, we hypothesized that the concept vector should first be tuned to better fit the corpus it is querying. This is similar to the idea that a *corpus-based* similarity thesaurus [40] is better than a general purpose one.

An ESA vector has two candidates for such tuning – the subset of concepts and the weights assigned to them. To check whether tuning should be performed for both of them, we ran the same tests as before, but with all query concept weights set to a uniform value. We found that this change hardly made any difference in performance, and this conclusion was also verified in similar tests in later experiments. Thus, we conclude that tuning the original concepts is useful only when altering the *set of concepts* to be used. We will focus on this in the next section.

Chapter 4

Selective ESA-Based Retrieval

We have shown that the basic ESA concept-based representation of a query or a document may be ambiguous and noisy, requiring tuning before it can be used efficiently. Before we propose tuning methods, we must decide where in the retrieval process the tuning should be applied. As the concept-based representation is used in both the document indexing and query processing stages, it would seem reasonable to suggest that tuning should also be done in both.

We chose, however, to focus on the query processing stage only. The main reason was that queries are much shorter than documents or even passages. For a longer text fragment, the generated concepts reinforce the main topics in the text and noise is restricted, whereas fragments such as short queries (typically 2-3 words in TREC Adhoc datasets) generate concepts that still contain much noise. In addition, tuning a document's representation during the indexing phase is problematic because it lacks the context provided by a given query, and a certain feature may be considered noise for one query but informative for another. Finally, changes in indexing parameters require reindexing, incurring extensive experimentation costs.

4.1 Feature Selection using Pseudo-Relevance Feedback

When ESA was applied to the text categorization task [17], it was vulnerable to the same problems we have just described. Nevertheless, the researchers overcame these problems by employing aggressive feature selection (FS). FS methods use labeled training examples to evaluate the utility of candidate features [23]. In text categorization, these examples are provided as part of the task data. In contrast, the IR task inherently lacks any labeled training data; hence applying FS to information retrieval will require finding an alternative method of evaluating the utility of features (concepts in our case).

For this purpose, we note that there exists also a supervised version of IR, called *relevance feedback* [43], where the user provides relevance judgments on an initial set of retrieved results. This feedback is then used to reformulate the query and retrieve an improved set of results. Relevance feedback can be extended to the unsupervised case, by assuming that the top ranked results (documents or passages) in the initial retrieved results are relevant [44]. This method is commonly referred to as *pseudo-relevance feedback* (PRF).

Inspired by PRF, we decided to use the results of keyword-based retrieval as a source for evaluation in our FS process. Our updated retrieval method will thus become two-phased, first performing *keyword-based* retrieval, then using its results to tune the query concepts and perform concept-based retrieval.

Next, we had to decide which subsets of the results are to be used. Most of the work on PRF used the top ranked documents or passages [42, 57] as pseudo-relevant documents (or *positive* examples). Some researchers chose to include also pseudo-non-relevant documents (or *negative* examples), by using the *bottom-ranked* documents [26]. We chose to use both positive and negative examples, as the initial query representation includes irrelevant concepts to be removed (for which we believe negative examples will be useful), in addition to missing relevant concepts (for which the positive examples alone are sufficient).

One may argue that, for the purpose of negative examples, randomly selected documents may make a better choice, in particular for queries with

many relevant documents. [47] analyzed a similar claim, when suggesting which documents should be used as non-relevant ones for learning a query profile for information filtering. They showed that sampling non-relevant documents from the "query zone" (meaning the set of non-relevant documents that are *similar enough* to the query) is better than sampling from the entire corpus (minus the relevant documents) when it comes to choosing features that are strong indicators of relevance.

Like the findings of [47], our early findings showed that using the bottom-ranking documents (a "query zone" equivalent) as non-relevant examples produced better results than using random documents. We also found early in our experimentation that keyword-based *passages* significantly outperformed full documents. This can be explained by the more coherent concepts produced by concise passages, similar to our findings in Section 3.3.2.

Following these findings, our algorithm will be using the top ranking keyword-based passages as positive examples, and the bottom ranking passages as negative examples. The next section will describe an algorithm for ESA-based retrieval that uses these pseudo-relevant examples to tune and select the query features.

4.2 Selective ESA-Based Retrieval Algorithm

Now that we have decided on a framework for evaluating features, let us describe the integration of FS into the general ESA-based retrieval algorithm. Since we chose to perform FS only on the query representation, the indexing algorithm is unchanged and remains as described in Figure 3.1, and we shall now elaborate on the revised retrieval algorithm, provided in Figure 4.1.

First, as in the non-selective algorithm, the textual query $\vec{q}$ is represented by an ESA concept vector $\vec{F_q}$. Then, the first n results ranked by keyword-based retrieval for $\vec{q}$ are fetched. The top k of these ($k \ll n$) are tagged as pseudo-relevant, or positive examples, and the bottom k are tagged as pseudo-non-relevant, or negative examples. Feature selection is then applied to these examples in order to select the best performing concepts in $\vec{F_q}$, resulting in a modified ESA vector $\vec{F_q'}$. Finally, concept-based retrieval is performed using $\vec{F_q'}$ and results are returned. The entire process is illustrated in Figure 4.2.

Given this generic algorithm and information on positive and negative

#Retrieve ESA concept-based results for query $\vec{q}$, cutoff concept vector at s;
 # select initial concepts based on k pseudo-relevant examples, taken from BOW
 # retrieval of depth n, to keep only fraction θ from initial set (where applicable)
Procedure SELECTIVE-ESA-RETRIEVAL($\vec{q}, s, k, n, \theta$)
$\vec{F_q} \leftarrow$ ESA($\vec{q}, s$)
$\langle d_1, .., d_n \rangle \leftarrow$ BOW-RETRIEVAL($\vec{q}, n$) *# ordered by relevance*
$\mathcal{D}_r \leftarrow \langle d_1, .., d_k \rangle$
$\mathcal{D}_{nr} \leftarrow \langle d_{n-k+1}, .., d_n \rangle$
$\vec{F'_q} \leftarrow$ FEATURES-SELECT($\vec{F_q}, \mathcal{D}_r, \mathcal{D}_{nr}, \theta$)
Return DOCSPASS-RETRIEVE($\vec{F'_q}$)

Figure 4.1: Selective ESA-based retrieval

examples, several actual FS methods can be suggested to implement the generic FEATURES-SELECT() step in the algorithm. In the following subsections we propose and experiment with three such FS methods.

4.2.1 Feature Selection using Information Gain

The first FS method uses each feature's *individual* utility to select a subset of the initial concept-based representation. This utility is measured by the information gained in separating the set of positive and negative examples [41]. Information gain (IG) was originally suggested in the context of a decision tree induction method for choosing which feature to branch on, but is also used extensively in feature selection [58]. For a feature f and a set $\mathcal{S}$ composed of positive and negative examples, the IG of f is calculated as the change in information entropy E when splitting $\mathcal{S}$ into subsets $\mathcal{S}_i$ according to their value of f:

$$IG(f, \mathcal{S}) = E(\mathcal{S}) - \sum_i E(\mathcal{S}_i) \cdot \frac{|\mathcal{S}_i|}{|\mathcal{S}|}$$

where $E(\mathcal{S})$ stands for the information entropy in a set $\mathcal{S}$. In our case, f is an ESA concept, and we define the value of f in each example to be the IR score of that example when f is used as the query. Since such feature values are continuous, they must be discretized in order to split them into subsets and calculate IG. Following [41], the feature values are discretized

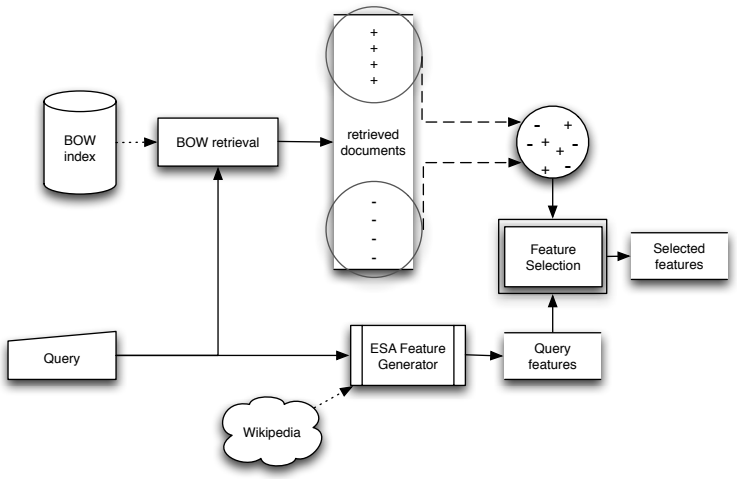

Figure 4.2: The PRF-based feature selection process

by calculating IG for every possible cutoff value, and using the best value as this feature's IG. The complete utility calculation is described in function $\mathcal{U}()$ in Figure 4.3. The function is generalized to calculate utility for a set of features as well, as some of our FS methods require.

We note that a feature that retrieves primarily negative examples is less useful for IR purposes. The scarcity of relevant documents and the random nature of non-relevant documents usually imply that very little information is expected to be added by such features. Our version of IG, shown as function IR-IG in Figure 4.3 and used by the utility function $\mathcal{U}()$, takes that into account by negating the result value when more negative examples are retrieved than positive ones. Negating (rather than setting to zero) also proves useful in producing a value that is easy to sort by, in case we have to select the "least-worst" features. One may argue that features with a large negative value may better be used in a negation retrieval clause (NOT operator), but our experiments showed no added value in doing so, which is probably explained by the incidental and anecdotal nature of those features.

The resulting IG feature selection method is shown in Figure 4.4. The procedure returns the best performing query features as measured by their IG values, cutting off at the requested level (θ).

#Calculate the utility for the feature set in $\vec{F}$, by calculating how well it separates
 # pseudo-positive examples $\mathcal{D}_r$ from pseudo-negative examples $\mathcal{D}_{nr}$
Function $\mathcal{U}(\vec{F}, \mathcal{D}_r, \mathcal{D}_{nr})$
$m \leftarrow |\mathcal{D}_r \cup \mathcal{D}_{nr}|$
$\langle d_1, .., d_m \rangle \leftarrow (\mathcal{D}_r \cup \mathcal{D}_{nr})$ sorted by their ranking in DocsPass-Retrieve($\vec{F}$)
best $\leftarrow \max_{i=1..m}$ (IR-IG($\mathcal{D}_r, \mathcal{D}_{nr}, \{d_1,..,d_i\}, \{d_{i+1},..,d_m\}$)
Return *best*

#Calculate the information gained by splitting the examples in $\mathcal{D}_r$ and $\mathcal{D}_{nr}$ into
 # the two subsets $\mathcal{S}_\oplus$ (predicted as relevant) and $\mathcal{S}_\ominus$ (predicted as non-relevant)
Function IR-IG($\mathcal{D}_r, \mathcal{D}_{nr}, \mathcal{S}_\oplus, \mathcal{S}_\ominus$)
$IG \leftarrow 1 - |\mathcal{S}_\oplus|/|\mathcal{S}_\oplus \cup \mathcal{S}_\ominus| \cdot$ Entropy($\mathcal{S}_\oplus$) $- |\mathcal{S}_\ominus|/|\mathcal{S}_\oplus \cup \mathcal{S}_\ominus| \cdot$ Entropy($\mathcal{S}_\ominus$)
If $|\mathcal{D}_r \cap \mathcal{S}_\oplus| < |\mathcal{D}_{nr} \cap \mathcal{S}_\oplus|$
$IG \leftarrow -IG$
Return IG

Figure 4.3: Utility calculation for a set of concepts to be used in IR

#Select a portion θ from the initial features $\vec{F}_q$, using positive examples set $\mathcal{D}_r$
 # and negative examples set $\mathcal{D}_{nr}$ to calculate IG for each feature
Procedure Features-Select-IG($\vec{F}_q, \mathcal{D}_r, \mathcal{D}_{nr}, \theta$)
$\langle f_1..f_{|\vec{F}_q|} \rangle \leftarrow$ sort $\vec{F}_q$ by $\mathcal{U}(\{f\}, \mathcal{D}_r, \mathcal{D}_{nr})$ in descending order
Return $\{f_1..f_{\lceil \theta \cdot |\vec{F}_q| \rceil}\}$

Figure 4.4: Selective ESA-based retrieval – IG selection method

4.2.2 Feature Selection using Incremental Information Gain

In the previous section, we described a selection method based on the IG value of each *individual* feature. In our case, however, these features are ultimately used as part of a complete set of query concepts, and dependency between the different features may imply that individual utility calculation is inaccurate. The *Incremental Information Gain* (IIG) method hypothesizes that feature utility would be better evaluated in the context of a full set of query features. Since examining all subsets of the initial feature set is exponential in the number of initial features and not computationally feasible, we perform a heuristic search in this space using our utility function $\mathcal{U}$ as the heuristic function.

#Filter original query features vector $\vec{F_q}$ by incrementally adding features that
 # improve or sustain best retrieval IG-based utility (calculated using $\mathcal{D}_r$ and $\mathcal{D}_{nr}$).
 # The parameter θ is ignored in this method.
Procedure FEATURES-SELECT-IIG($\vec{F_q}, \mathcal{D}_r, \mathcal{D}_{nr}, \theta$)
$\langle f_1..f_{|\vec{F_q}|} \rangle \leftarrow$ sort $\vec{F_q}$ by $\mathcal{U}(\{f\}, \mathcal{D}_r, \mathcal{D}_{nr})$ in descending order
$\mathcal{F}'_q \leftarrow \{\}$
For i **from** 1 **to** $|\vec{F_q}|$
If $\mathcal{U}(\mathcal{F}'_q \cup \{f_i\}, \mathcal{D}_r, \mathcal{D}_{nr}) \geq \mathcal{U}(\mathcal{F}'_q, \mathcal{D}_r, \mathcal{D}_{nr})$
$\mathcal{F}'_q \leftarrow \mathcal{F}'_q \cup \{f_i\}$
Return $\mathcal{F}'_q$

Figure 4.5: Selective ESA-Based Retrieval - IIG selection method using forward-selection

The IIG method builds the representation incrementally, using forward selection or backward elimination [27]. Features are first sorted by their individual IG value, and the candidate query set is an empty one (or the full one, for backward-elimination). Then, in each iteration a feature is added to the candidate set (or removed, for backward-elimination) if this step does not degrade[1] current pseudo-relevance based performance, or discarded otherwise. When all features have been evaluated, the algorithm terminates

[1]This condition implies that for forward-selection we will keep redundant features, whereas for backward-elimination we will remove them. We elaborate on this in the results section.

and returns the selected features. In addition to the advantage of evaluating the feature in the context of other features, this method also has the advantage of not requiring a predefined selection level, thus removing one parameter from the system.

Figure 4.5 shows the IIG selection method, when using forward-selection. For backward-elimination, the algorithm will begin with the full feature set, and in each iteration attempt to eliminate the lowest-performing feature, choosing to keep it if its removal harms performance.

4.2.3 Feature Selection using Rocchio's Vector

In the two previously described FS methods, the set of candidate features were those generated for the query by the ESA feature generator, $\vec{F_q} = ESA(\vec{q}, s)$. However, the extremely short queries (1-3 words in the datasets we used) may not suffice to generate and assign high weight to important concepts.

Consider query 415 in TREC-8, "drugs, Golden Triangle." This query refers to an area in southeast Asia that is known for illicit opium production, but since no such single explicit concept existed in our ESA model, the query's top concepts were related to other "golden triangle" meanings, and relevant topic-related concepts were not considered. Employing FS on the generated concepts was naturally not helpful, as the initial candidate set's coverage was not sufficient.

Yet, our ESA model does include other features that could represent the correct "golden triangle" using other concepts, such as Illegal drug trade, Opium, Myanmar and Laos (two countries located in this triangle). Such ESA concepts could be generated from texts discussing the correct query interpretation. Since the top retrieved documents for the *keyword-based* query are expected to be such texts, we may use them to try and compensate for the inaccurate query concepts. Hence, we would like to generate and use these concepts as additional concepts in the set of candidate features to be selected.

We thus propose a new FS method where the augmented set of candidate features is $\vec{F_q} = ESA(\vec{q}, s) \bigcup \{ESA(\vec{d}, s) \mid \vec{d} \in D_r\}$. Now we need to evaluate and select features from this set. Using IG to evaluate how well each feature separates top-ranking documents from bottom-ranking ones

 # use positive examples set $\mathcal{D}_r$ and negative examples set $\mathcal{D}_{nr}$,
 # to reformulate initial query features vector $\vec{F}_q$,
 # and return θ strongest fraction of resulting vector
Procedure FEATURES-SELECT-RV($\vec{F}_q, \mathcal{D}_r, \mathcal{D}_{nr}, \theta$)
$\vec{F}'_q \leftarrow \vec{F}_q + \frac{1}{|D_r|}\sum_{\vec{d}\in D_r} ESA(\vec{d}, s) - \frac{1}{|D_{nr}|}\sum_{\vec{d}\in D_{nr}} ESA(\vec{d}, s)$
$\langle f_1..f_{|\vec{F}'_q|}\rangle \leftarrow$ sort $\vec{F}'_q$ by weight of f in descending order
Return $\{f_1..f_{\lceil\theta\cdot|\vec{F}'_q|\rceil}\}$

Figure 4.6: Selective ESA-based retrieval – RV selection method

is not sound, as the additional features were already taken from the top ranking documents. Instead, we will use the weights of each feature in each document in the sets of positive and negative examples, average these values into a combined weight and use the results to select features.

We calculate the features' weights based on Rocchio's algorithm for relevance feedback [43]. Each feature receives a weight that is the sum of its weights in the original query and in the positive example documents, and then its weights in the negative example documents are subtracted. Finally, the strongest features are kept and the rest discarded. The pseudocode for applying the RV method is provided in Figure 4.6.

4.3 Empirical Evaluation

This section describes experiments carried out using selective ESA-based retrieval with each of the selection methods, and a comparative analysis of the results.

4.3.1 Methodology

We continue using the experimental framework described in Section 3.3.1, and evaluate each suggested selection method with various system parameter settings. The following parameters have been fixed to a predefined value in all these experiments: s, the concept vector cutoff, has been set to 50; and n, the BOW retrieval depth for pseudo-relevance, has been set to the first 1000 results. The system parameters we will be experimenting with are k, the

pseudo-relevant result set size, and θ, the feature selection aggressiveness level (where applicable).

To further assess the value of feature selection in itself, we also experimented with a fourth, *random* method, which randomly selects a subset of features of the required size (as defined by θ) from the original representation, regardless of the provided examples. We used this method to reject the hypothesis that an observed improvement in performance may solely or partly be attributed to the use of a smaller subset of the original features rather than the specific features selected.

4.3.2 IG Method Results

The IG method has two primary parameters: the number of pseudo-relevant examples (k) and the selection level (θ). Figure 4.7 shows retrieval performance (averaged over all queries in each dataset) as a function of θ for several values of k, compared with a baseline that performs no FS at all. Both datasets show similar behavior, with FS performance consistently improving as selection level increases, peaking at $\theta = 20\%$ (which implies retaining 10 out of the initial 50 features). More aggressive selection is already damaging, probably as the result of removing useful features along with non-relevant ones.

Figure 4.8 shows the same experiment from a different perspective, with performance as a function of k for several values of θ. The number of examples used seems to influence performance less than selection level, except when too few examples are used ($k = 5$), resulting in insufficient information for IG to be reliable. Nevertheless, adding more and more examples degrades rather than improves performance. This may be attributed to the decrease in actual relevance of the pseudo-relevant examples, when taken from lower rank positions.

4.3.3 IIG Method Results

The IIG method requires only one parameter to be set, the size of the positive/negative example set (k). In addition, the algorithm may be run in forward-selection or in backward-elimination mode. Figure 4.9 shows retrieval results for different values of k in both modes, compared with results of the initial baseline query.

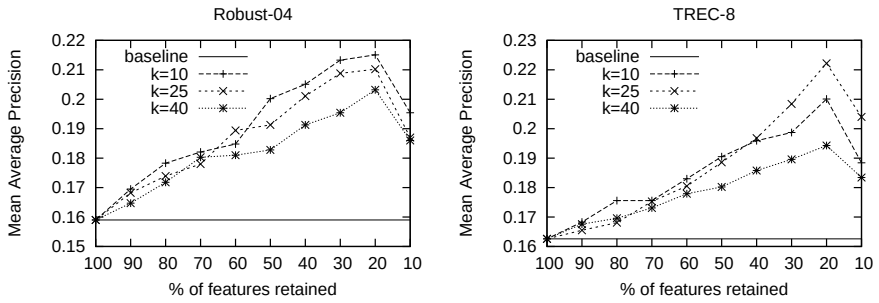

Figure 4.7: Concept-based performance as a function of a fraction of the concepts selected (θ), IG method

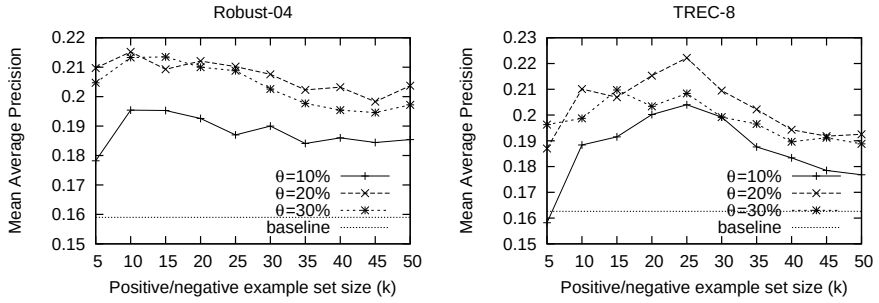

Figure 4.8: Concept-based performance as a function of the number of pseudo-relevant examples (k), IG method

In both datasets, the IIG method shows consistent improvement over the performance of the baseline. The results also show the forward-selection approach consistently outperforming the backward-elimination approach. One reason we found for this was the inherent filtering of redundant features in backward elimination. If a certain query has two highly informative but similar features, it is quite possible that each alone will be sufficient to perfectly separate the positive from negative examples. Then, backward elimination will eliminate one of them, as its removal does not degrade performance, although in a full corpus retrieval, that additional feature could have contributed to the query's performance. This may also explain why the difference between forward selection and backward elimination is greatest when very few examples are used.

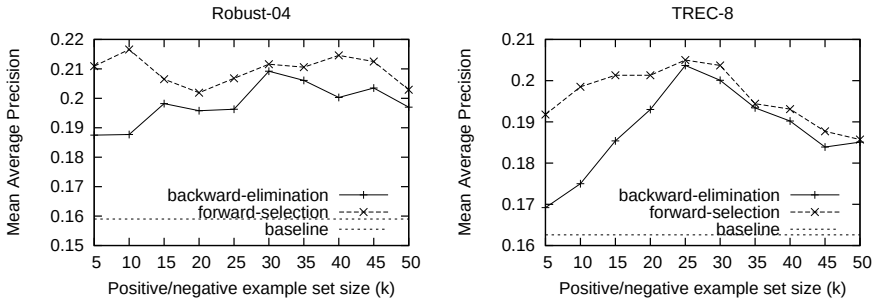

Figure 4.9: Concept-based performance as a function of the number of pseudo-relevant examples (k), IIG method

We also experimented with another variation of the IIG method, where the weights of each examined feature were recalculated in each iteration. In this version, the next feature to add will be the one to maximize the local value $\mathcal{U}(\mathcal{F}'_q \cup \{f\})$ rather than the global $\mathcal{U}(\{f\})$ used in the algorithm described in Figure 4.5. Despite this version being more in line with common practice hill climbing implementations, it performed well below the global one. We suspect this is due to the inaccurate nature of the example documents, which increase the chance for local maxima.

4.3.4 RV Method Results

The RV method, like IG, requires setting two parameters, k and θ. Like the graphs in the previous sections, the graphs in Figures 4.10 and 4.11 show the impact of these parameters on the system's performance. But whereas with the IG method the query reverts to the original query at $\theta = 100\%$, this is not the case with the RV method. Even without any selection, the query changes as a result of adding the features generated from the positive example documents and of the reweighting step.

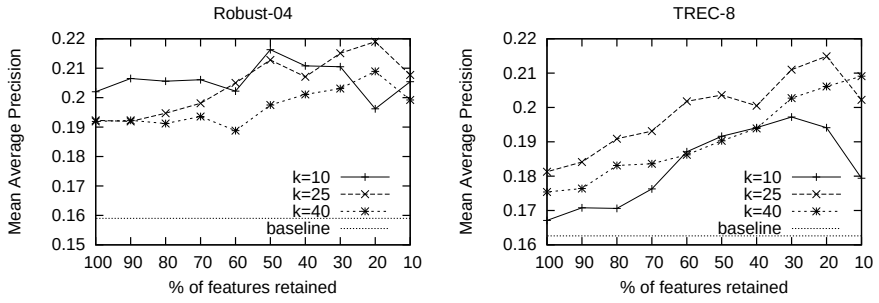

Figure 4.10: Concept-based performance as a function of a fraction of the concepts selected (θ), RV method

Figure 4.10 shows that even without selection, performance is better than the baseline, and that the improvement generally increases with the selection level (except for the very high selection levels). Figure 4.11 shows that using a very small set of examples ($k = 5$) yields poor results, with performance improving and stabilizing as more examples are provided. Once performance stabilizes. adding further examples does not seem to make much difference. The impact of FS is also clearly demonstrated, with the $\theta = 100\%$ curve mostly lower than the highly selective curves.

4.3.5 Random Selection Results

We replaced the PRF-based selection process with a random one. A subset of the required size (determined by the parameter θ) was randomly sampled from the initial query features set for each query, and retrieval results for these randomized concept-based queries were evaluated. This process was

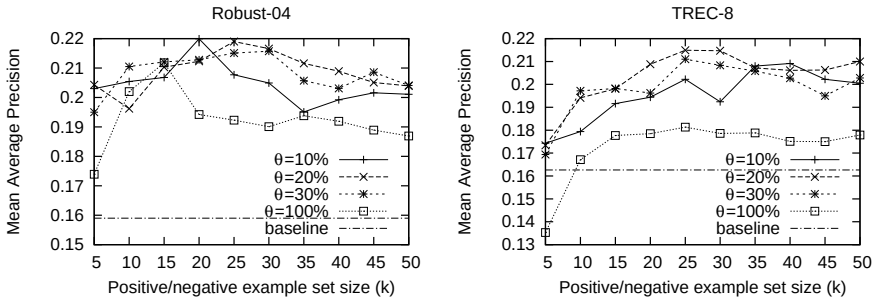

Figure 4.11: Concept-based performance as a function of a fraction of the number of pseudo-relevant examples (k), RV method

then repeated 10 times (for each choice of θ). The parameter k was irrelevant for these experiments, as the examples were not used in any way.

The results in Figure 4.12 show a continuous *decrease* in performance as more features are randomly removed from the initial set. This clearly indicates that the improvement shown by previous methods must be attributed to the specific set of features chosen, rather than just the act of using a smaller set of features.

4.3.6 Parameter Tuning through Training

All selection methods shown in this section rely on one or two system parameters, whose values may have a significant impact on system performance. These parameters can be tuned if a set of queries is provided with relevance judgments on result documents. We used a third dataset, TREC-7 [54], which shares the same corpus as TREC-8 and Robust-04 but has a different set of queries, to perform parameter tuning.

We operated the system on TREC-7 queries with the three proposed FS methods and varied the parameter value ranges. The resulting best performance values obtained were: for IG FS $\langle k = 10, \theta = 30\% \rangle$, for IIG FS $\langle \text{forward-selection}, k = 10 \rangle$ and for RV FS $\langle k = 35, \theta = 20\% \rangle$. All of these values fall well within the top performing value ranges (though not always the peak values) in TREC-8 and Robust-04. This result, coupled with the similarity of the system's performance graphs over the TREC-8 and Robust-04 datasets, suggests that relatively consistent system behavior

can be expected, and ESA-based systems may be tuned on one set of queries and then used on other test sets.

4.4 Analysis

Having evaluated each of the suggested FS methods, let us examine the results in greater depth. We have demonstrated that feature selection on the query concept vector is effective in obtaining better retrieval results, and that this improvement is not the result of merely using a smaller set of concepts. Now let us compare the effectiveness of each method, in order to draw some general conclusions as to what scenario they may be best suited for.

The IG method exhibits good peak behavior, but it seems to be highly sensitive to the chosen selection level θ. Tuning the system parameters using training data, if available, may significantly alleviate this problem, as shown by the tuning experiment we conducted.

The IIG forward-selection method appears to perform better than backward-elimination. This method requires tuning only a single parameter – the number of examples to be used. It would therefore be the preferred choice when no training data is available. Its performance, though, is slightly lower than IG, and it is still quite sensitive to the k parameter value.

The RV method performs slightly worse than IG for small example set sizes, probably due to its overdependence on the quality of these examples (as they are a source of generating features, not just filtering harmful ones). For larger example sets (in our case, $k > 15$), it performs comparably to the IG method. In addition, the RV method appears to be more *robust* than the other two, in that it yields overall good results for a broader range of parameter settings, rather than a pinpointed peak, and therefore will depend less on accurate parameter tuning.

Let us now revisit the Estonian economy example from Section 3.3.3. The revised query, after being processed by the RV method (as an example), is:

Economy of Estonia
Monetary policy
Estonia
Euro

Economy of Europe
Neoliberalism
Tiit Vähi
Prime Minister of Estonia
Eurozone
Nordic countries

The noisy sports-related concepts that appeared in the initial features are now filtered out of the query, as they appear very rarely (if at all) in the concepts of both sets of positive and negative examples. Other concepts that may seem relevant at first, such as Estonia and Baltic Sea, are filtered out for being too broad, appearing *frequently* in the concepts of both example sets. Concepts that are highly relevant to Estonia's economy, such as Economy of Estonia, Tiit Vähi and Eurozone, are retained in the top positions, while other relevant ones percolate upwards. In addition, the RV method also added Neoliberalism, an economy-related concept relevant to Estonia's economy that was not included in the original query concepts but appeared frequently in the concepts of positive examples.

To summarize, Figure 4.13 shows the improvement over the baseline for the three methods for $\theta = 20\%$. On the basis of this figure, we can state that adding FS to ESA concept-based retrieval can significantly improve retrieval results, with improvement of up to 40% over the non-selective ESA baseline in both datasets.

Nevertheless, even with this significant improvement, retrieval performance still stands at just over 85% of our BOW baseline. We believe that an inherent bias in the evaluation methodology may contribute to this low measured performance, and we will elaborate on this issue in Section 7, but for now we want to find ways to further improve the result. In the following section we show how information already available to the system can be further leveraged to produce our final and best performing algorithm.

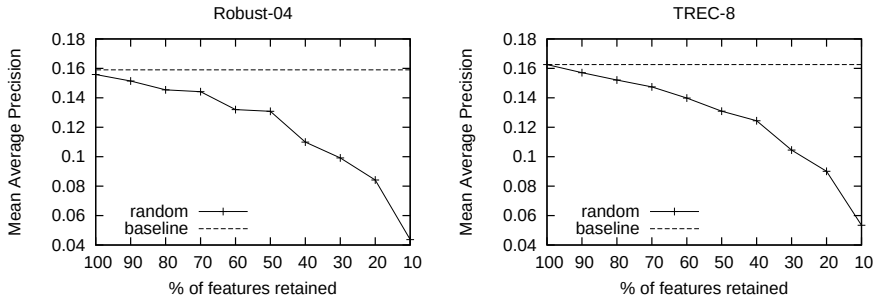

Figure 4.12: Performance of random selection method, averaged over 10 runs each

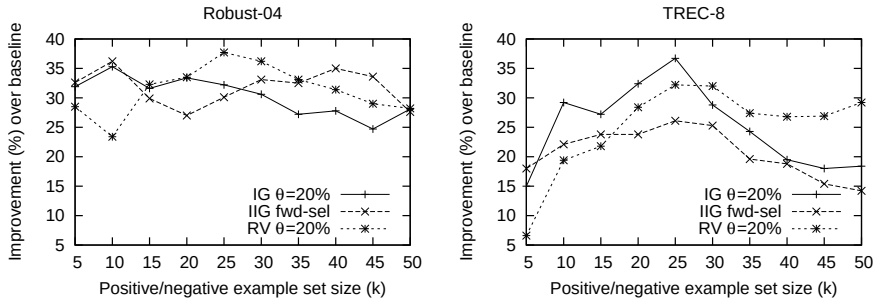

Figure 4.13: Best performing runs for all three FS methods

Chapter 5

Fused Selective ESA-Based Retrieval

A large body of research [13, 30, 50, 8] shows that *combining* (also known as 'fusion' of) retrieval methods may improve final results. Fusion of ranking approaches is known to achieve best results when the methods to be combined are substantially different in their approach [30]. With the significant difference between BOW and ESA representations, we expect that combining them will also yield better results. This idea is further reinforced by the findings of [16] in applying ESA to text categorization, which showed that augmenting the BOW representation with ESA concepts outperforms each individual representation alone.

5.1 Fused Selective ESA-Based Retrieval Algorithm

A survey of combining approaches can be found in [8]. In our study we use the simple, widely used model of Linear Combination [50], where document scores are weighted sums of the scores assigned by the individual retrieval methods to be fused, with weighting determined using training data. Before summing, document scores are normalized to account for the different ranges in score values, as suggested by [30].

Once both retrieval results (concept-based and keyword-based) are normalized, document scores are then weighted and summed using the weight factor w, provided as an additional parameter. The pseudocode for this algorithm is described in Figure 5.1. The value for this parameter can be obtained using parameter tuning on a dataset with relevance judgments.

#Retrieve ESA concept-based results fused with BOW keyword-based results for query $\vec{q}$,
 # cutoff concept vector at s and select using k pseudo-relevant examples, all retrieval
 # is to depth n. Scores are a weighted sum using the parameter w as the weight.
Procedure FUSED-SELECTIVE-ESA-RETRIEVAL($\vec{q}, s, k, n, w$)
$\mathcal{B} \leftarrow$ BOW-RETRIEVAL($\vec{q}, n$)
$\mathcal{E} \leftarrow$ SELECTIVE-ESA-RETRIEVAL($\vec{q}, s, k, n$)
$\mathcal{D} \leftarrow \mathcal{B} \bigcup \mathcal{E}$
#calculate normalized score in both retrieval modes (zero if not retrieved)
Foreach $d \in \mathcal{D}$
$normScoreBOW(d) \leftarrow \frac{score(d,\mathcal{B}) - \min_{b \in \mathcal{B}}(score(b,\mathcal{B}))}{\max_{b \in \mathcal{B}}(score(b,\mathcal{B})) - \min_{b \in \mathcal{B}}(score(b,\mathcal{B}))}$
$normScoreESA(d) \leftarrow \frac{score(d,\mathcal{E}) - \min_{e \in \mathcal{E}}(score(e,\mathcal{E}))}{\max_{e \in \mathcal{E}}(score(e,\mathcal{E})) - \min_{e \in \mathcal{E}}(score(e,\mathcal{E}))}$
$score(d) \leftarrow w \cdot normScoreESA(d) + (1 - w) \cdot normScoreBOW(d)$
$\langle d_1 \dots d_{|\mathcal{D}|} \rangle \leftarrow$ sort $\mathcal{D}$ by $score(d)$ in descending order
Return $\langle d_1 \dots d_n \rangle$

Figure 5.1: Fused selective ESA-based retrieval – the MORAG algorithm

5.2 The Morag System

Let us now recap the entire resulting system, which we named MORAG[1], as illustrated in Figure 5.2. First, an ESA model is built from Wikipedia or another source, as described in [17]. During the indexing stage, MORAG indexes the corpus in both BOW and ESA representations. Then, at retrieval time, the BOW query is submitted; its results are kept for the fusion phase and also fed into the FS module, together with the ESA query representation. After FS is complete, the resulting features are used to perform a concept-based retrieval, and the results of the concept-based and keyword-based retrieval runs are fused to produce the final MORAG results.

Note that in our implementation of MORAG we have used the same BOW subsystem for both purposes: generating pseudo-relevant examples, and fusion to concept-based results. However, other implementations using different BOW retrieval systems for each of these purposes are also possible.

[1] Morag is the Hebrew word for *flail*, an agricultural tool used to separate grain from chaff.

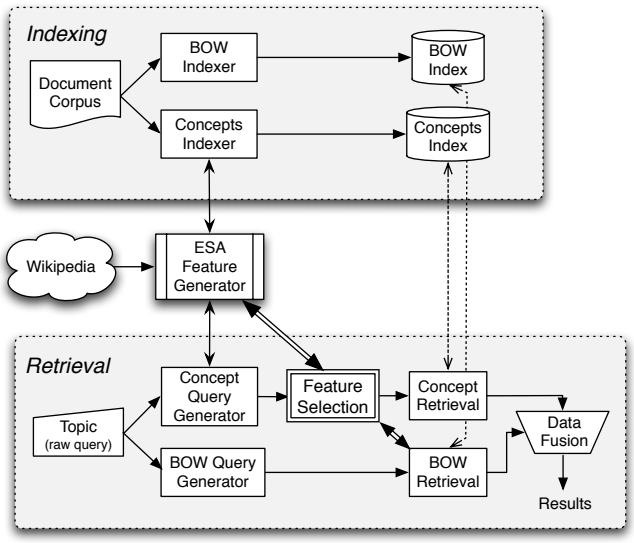

Figure 5.2: The MORAG solution architecture

5.3 Empirical Evaluation

We ran a set of experiments to evaluate the performance of the MORAG system, and to analyze its robustness and further potential.

In addition, we evaluated MORAG in combination with and in comparison to top performing systems in TREC-8. As [3] recently pointed out, it is not sufficient for IR researchers to show improvement over their own baseline, rather they should strive to show that their method can improve over systems that are already highly effective. We will show that our method is indeed capable of doing that.

5.3.1 Methodology

The experimental methodology generally follows that of the previous section. Specifically, in this algorithm, we also need to tune the value for the parameter w. We used the TREC-7 dataset for this purpose too, selecting the parameter value that maximized the performance of MORAG on TREC-7, which was found to be $w = 0.5$ for the combination of ESA and Xapian BOW.

Dataset	*Baseline*	MORAG − *IG* *tuned*	MORAG − *IIG* *tuned*	MORAG − *RV* *tuned*	MORAG *optimal*
TREC-8	0.2481	0.2864(+15.4%)	0.2734(+10.2%)	0.2888(+16.4%)	0.2947(+18.8%)
Robust-04	0.2622	0.2914(+11.1%)	0.2923(+11.5%)	0.2879(+9.8%)	0.3010(+14.8%)

Table 5.1: Performance of MORAG using tuned parameter values and optimal parameter values. Improvement percentage over baseline is shown in parentheses next to each result.

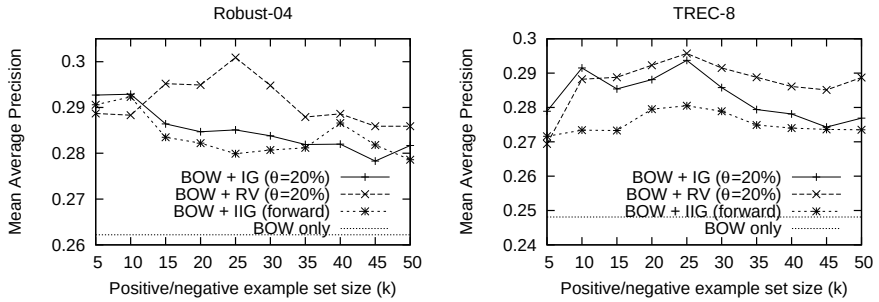

Figure 5.3: MORAG performance as a function of a fraction of the number of pseudo-relevant examples (k), all methods

5.3.2 Morag Results

Table 5.1 shows results for both TREC-8 and Robust-04 datasets for all three FS methods, with parameters tuned on the TREC-7 dataset. The last column shows the system's performance with optimal choice of parameters, as an indicator of what further improvement can be achieved by better parameter tuning.

The results show an impressive improvement over the BOW baseline, for all FS methods. Parameter tuning yields reasonable results: 55%-85% of the optimal performance. We checked the statistical significance of the results using a paired two-tailed t-test, and all the results were significant at $p > 0.95$.

Figure 5.3 compares performance for the different selection methods in MORAG, for various values of the parameter k, assuming that the parameter θ is easier to optimize due to its peak behavior (or its irrelevance for IIG). The results show the RV method achieves best results, for sufficiently large values of k.

Figure 5.4 shows results for one specific choice of selection method and level, comparing the performance of the fused system with that of its components. The graph demonstrates how fusion with ESA-based results improves the system's per-

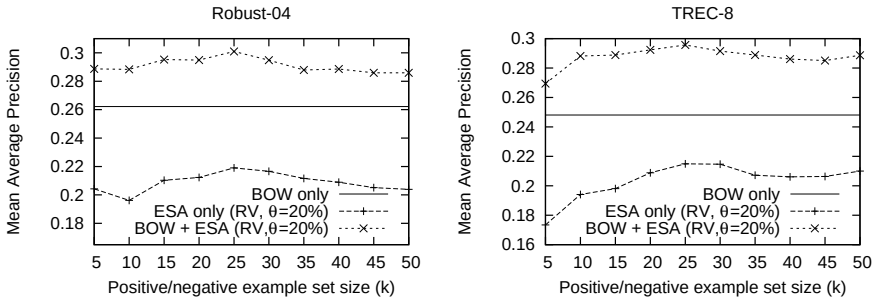

Figure 5.4: Comparison of fused results with results of each fused subsystem on its own (for a single choice of FS method and selection level)

formance by an increment that is correlated with the ESA system's performance, as expected. Note that despite the relatively low performance of the ESA run, fusion still yields good improvement. Similar behavior is also observed for the other methods and selection level values.

5.3.3 Fusion with Alternative BOW Subsystems

The previous experiments were carried out using our choice of an experimental BOW system (Xapian). However, since Morag is modular, it can be used with any other BOW component, and we were interested in assessing the system's robustness over different (and better performing) BOW systems.

We used two additional effective and common retrieval approaches implemented in the Lemur toolkit[2]: a TF.IDF-weighted vector space model with pseudo relevance feedback (we denote this run FB-TFIDF), and a language model based on KL-divergence using Dirichlet prior smoothing (denoted LM-KL-DIR). We used the "out of the box" Lemur implementations with default parameter values, and set the Morag-specific parameters for these systems by parameter tuning on TREC-7.

In addition, we wanted to use systems that achieved the highest results in the original TREC runs. Normally, such experimentation is not feasible, since these systems (or their exact detailed implementation) are usually not available, and evaluations of this kind are not common in IR. However, since Morag performs fusion on the result-set level (rather than change the core ranking functions), such comparisons are possible in our case using only the target systems' output. TREC provides access to past participants' raw results, and we used this data as additional BOW systems.

[2]http://www.lemurproject.org/

When determining which TREC systems are best to compare with, we searched for those that employed standard BOW approaches, were among the top performing on the evaluated datasets, and that participated in TREC-7 (with no major internal changes) so that we could also perform parameter tuning using their TREC-7 results. We could not find candidates in the Robust-04 dataset that were good enough for this comparison; hence, we will show results only on the TREC-8 dataset.

Note that the BOW system is used twice in Morag – once as a source for PRF, and once for fusing the results. However, since relevance feedback in Morag is *passage-based*, and the system outputs we had access to were *document-based* results, we still had to use our own BOW baseline for the PRF stage.

We used BOW results from the Okapi [42], PIRCS [29] and AT&T [46] teams, which were 3 of the top performing systems of TREC-8 participants using short queries. The Okapi and AT&T teams augmented standard BOW retrieval with extensive query expansion methods based on PRF, while the PIRCS team used a system that combined different BOW retrieval models (probabilistic and language modeling). As stated earlier, our relevance feedback utilized Xapian passage-based results for all runs, and the ESA FS method used in these experiments was IG. All three teams stated in their publications that their system was virtually the same as that used in TREC-7; hence we take the parameter tuning on TREC-7 to be valid for these systems as well.

Table 5.2 shows the improvement gained by using each of these systems as the BOW component in Morag. The third column shows results when fusing with tuned parameter values as described above, while the fourth column shows results for optimal parameter values. We evaluated these results for statistical significance as well, and significant results are marked in boldface.

These results demonstrate that improvement can also be achieved with top performing BOW systems, although the added value of the fusion was lower in those cases. This is understandable, given the current relatively low performance of ESA retrieval alone, and considering that successful fusion is known to require the fused systems to have comparable performance levels [8].

5.3.4 Comparison to Fusion of BOW Systems

Fusing results from two retrieval systems is known to be a potential source of improvement in itself [8], regardless of the underlying text representations. To assess the true contribution of ESA concepts to the results shown thus far, we wanted to measure what portion of the improvement gained by Morag can be attributed solely to the act of fusing results. To do so, we compared the improvement attained by Morag with that attained by fusing the baseline BOW results with results of another BOW system whose measured performance is similar to that of our concept-based retrieval subsystem.

BOW system	*Baseline*	MORAG (*tuned*)	MORAG (*optimal*)
Xapian	0.2481	**0.2864 (15.4%)**	**0.2947 (18.8%)**
LM-KL-DIR	0.2498	**0.2877 (15.2%)**	**0.2924 (17.1%)**
FB-TFIDF	0.2697	0.2829(4.9%)	**0.2951 (9.4%)**
Okapi	0.2787	**0.3042 (9.1%)**	**0.3065 (10.0%)**
AT&T	0.2853	0.2977(4.3%)	**0.3096 (8.5%)**
PIRCS	0.3063	0.3211(4.8%)	**0.3239 (5.7%)**

Table 5.2: TREC-8 results for MORAG with several BOW baselines, using tuned parameter values and optimal parameter values. Improvement percentage is provided in parentheses. Statistically significant results are marked in boldface.

We compared optimal results for MORAG with optimal-w results of fusion with several other TREC-8 participants who applied the BOW approach and used short queries: RMIT [14], ACSys [24] and INQUERY [1]. These three system runs had a comparable or slightly higher MAP score than our ESA-based run, and fusing them with each of the BOW systems in the table provides an indication of the value of fusion itself. We used optimal rather than tuned w values, since only one of these participant groups (INQUERY) stated that no changes were made between TREC-7 and TREC-8, and hence training on TREC-7 was not sound.

Table 5.3 shows the results of these experiments. For comparison, the last column lists the optimal MORAG improvements again. The obtained results are much poorer than MORAG's and most are not statistically significant, despite being produced by fusion with systems that perform slightly better than our ESA retrieval method. This indicates that the improvement in the previous section cannot be attributed solely to fusion, and demonstrates the added value in the concept-based retrieval component of MORAG. This finding is also in line with [30], who posited that combining retrieval approaches works best when the representation and weighting schemes differ significantly.

5.3.5 Additional Measures and Analysis

We have shown in the previous sections that fusion with ESA concept-based retrieval produces better results than fusing with BOW systems. We now try to better understand why this is so.

Table 5.4 shows additional IR measures for several of the tested BOW systems, listing measured values for the baseline run of each system (first line), for the MORAG run using that system (second line) and for a run that fuses with another

BOW system	*+RMIT* (MAP=0.2236)	*+ACSys* (MAP=0.2309)	*+INQUERY* (MAP=0.2325)	MORAG (MAP=0.2223)
Xapian	**0.2524 (+1.7%)**	**0.2569 (+3.5%)**	**0.2586 (+4.2%)**	**0.2947 (+18.8%)**
Okapi	**0.2921 (+4.8%)**	0.2882(+3.4%)	**0.2903 (+4.1%)**	**0.3065 (+10.0%)**
AT&T	0.2943(+3.2%)	0.2933(+2.8%)	0.2897(+1.5%)	**0.3096 (+8.5%)**
PIRCS	0.3086(+0.8%)	0.3068(+0.1%)	0.3075(+0.4%)	**0.3239 (+5.7%)**

Table 5.3: Comparison of MORAG TREC-8 results (optimal parameter values) with TREC-8 results of BOW-BOW fusion (optimal w values). Statistically significant results are in boldface.

BOW system (fourth line). In this latter run, for Xapian and Okapi we used the best performing fusion in Table 5.3, while the two Lemur runs were fused with each other.

Examining the "P@5", "P@10" and "relevant retrieved" columns, we observe that the improvement in MAP demonstrated by MORAG is not to be attributed primarily to an improvement mainly in recall or mainly in precision - both measures are substantially improved. To further assess the improvement in recall we have also measured the overlap in relevant documents retrieved between each pair of fused systems ("overlap of relevant" column). Little overlap between the systems means that there is more chance that each system contributes new relevant documents to the pool, thus higher chances for higher overall recall. However, the Lemur fusion run (LM-KL-DIR w/FB-TFIDF), where 2922 of the final 3124 relevant documents are shared between the two fused runs and yet the overall recall is higher than MORAG's and nevertheless the final MAP is lower, demonstrates that other factors need to be examined to get the full picture.

The "non-rel retrieved" column measures the number of documents retrieved by each system, that were judged by TREC assessors to be *not relevant* for the dataset's queries. The results in this column indicate that MORAG consistently *reduces* the number of non-relevant documents retrieved, whereas the BOW fusion usually increases this number. This can be explained by the different ranking approach taken by a concept-based method: many non-relevant documents retrieved by a keyword-based approach may include the query terms in a high frequency but are not related to the query. Other keyword-based systems, ranking by similar principles, are likely to rank these documents high as well and reinforce these false positives, whereas a concept-based approach, ranking by conceptual similarity, is more likely to rank them low. This hypothesis is further reinforced by the "overlap of non-rel" column, where we explicitly quantify this overlap.

If we now revisit the Lemur fusion run, we'll notice that the two fused Lemur

methods have not only a high overlap in relevant documents, but also a significantly high overlap in non-relevant documents. Such a high overlap implies that non-relevant documents are reinforced too, thus hurting the overall precision despite the substantial improvement in recall. This low result is despite the fact that the fused systems perform well individually and use quite different ranking approaches.

Finally, we point at a third group of documents worth examining – the un-judged documents. The "pooling" method used in the TREC methodology [55] implies that only a small fraction of the corpus is evaluated for relevance by the human assessors, and any un-judged documents are then assumed to be non-relevant. This approach was found to work well when comparing the *relative* performance of IR systems. However, research has shown that the use of pooling could discriminate against a new method that is based on *novel* principles [61], and it has been recommended that researchers consider the number of un-judged documents being fetched as an indication that performance is probably being underestimated. Following this recommendation, we found that our concept-based runs retrieved almost 40% more un-judged documents than an average BOW system (about 35000 documents compared to about 25000 in the evaluated BOW systems). Hence, there is reason to suspect that the true performance of Morag may be even higher than the reported results, since some of these un-judged documents may well be relevant documents that could not be detected by any of the previous BOW approaches.

5.3.6 The Impact of Using More Relevant Examples

In this research, we have used the top and bottom ranked documents (in BOW retrieval) as positive and negative examples in the feature selection process. Naturally, these pseudo-relevant examples are a practical compromise, as they are *assumed* to be relevant (or non-relevant) but may not be so in practice. Ideally, we would prefer to use only documents indicated as relevant or non-relevant by the user. In considering this compromise, we were interested in learning more about the possible improvement to be gained by using better examples, and conducted additional experiments relying on TREC's human relevance judgments as "oracle" knowledge.

The retrieval process in these experiments was similar to that described in Section 4, except for the choosing of positive examples, for which we added a step of iterating through the top retrieved documents and selecting only those marked as relevant for this query in the TREC relevance judgments. Thus, the k positive examples were chosen from a larger subset of top documents, and were guaranteed to be relevant. Negative examples are chosen as before, since *relevant* documents are very unlikely to appear in the bottom-ranked documents, and it is even less likely that the bottom ranked documents will be judged at all. We then compare the results with those using standard pseudo-relevant positive examples.

BOW system	*MAP*	*P*@5	*P*@10	relevant retrieved	overlap of relevant	non-rel retrieved	overlap of non-rel
Xapian	0.2481	0.484	0.472	2735		20106	
*w/*MORAG	0.2864	0.552	0.478	3062	1824	19400	7115
	(+15.4%)	(+14.0%)	(+1.3%)	(+12.0%)		(-3.5%)	
*w/inq*601	0.2586	0.484	0.462	2907	2299	21436	15180
	(+4.2%)	(0.0%)	(-2.1%)	(+6.3%)		(+6.6%)	
Okapi	0.2787	0.552	0.488	3013		21271	
*w/*MORAG	0.3042	0.580	0.522	3168	2161	20410	8531
	(+9.1%)	(+5.1%)	(+7.0%)	(+5.1%)		(-4.0%)	
w/RMIT	0.2921	0.536	0.474	3095	2370	22878	13790
	(+4.8%)	(-2.9%)	(-9.2%)	(-2.3%)		(+12.1%)	
LM-KL-DIR	0.2498	0.468	0.442	2857		22048	
*w/*MORAG	0.2877	0.552	0.506	3087	2042	20553	7759
	(+15.2%)	(+17.9%)	(+14.4%)	(+8.1%)		(-6.8%)	
*w/*FB-TFIDF	0.2717	0.488	0.444	3124	2922	22450	17012
	(+8.8%)	(+4.3%)	(+0.5%)	(+9.3%)		(+1.8%)	

Table 5.4: Additional IR evaluation measures for TREC-8 results using several BOW baselines. BOW systems are fused with concept-based retrieval (using MORAG) and with another BOW system for comparison.

Figure 5.5 shows results for the TREC-8 dataset using the IG FS method, with and without "oracle" relevance knowledge in choosing positive examples. The results indicate that using verified relevant documents as positive examples indeed improves performance by about 10%-15%. In addition, using more examples does not degrade performance as it did with pseudo-relevant examples (see, for example, Figure 4.8), reinforcing our earlier assumption that the decrease in performance was due to the decreasing relevance of lower ranking documents. This result implies that there is value in more refined methods of choosing pseudo-relevant examples, which could be the subject of future work.

5.3.7 Estimating Optimal FS Performance

The ESA-based performance was shown in Section 4.3 to depend directly on the choice of subset: a better selection process yielded better performance. It will be safe to assume that further research could derive even better FS methods than those described, and consequently better overall performance. We believe, therefore, that it would be worthwhile to estimate how much further improvement can be expected by employing Morag with better feature selection methods.

In this final experiment, we iterated across all possible subsets of each query's initial features, and instead of using the described FS methods, we evaluated the subsets with relevance ("oracle") knowledge to find the one that gives optimal performance. Naturally, this process cannot be applied in a real-life scenario, but its results indicate the improvement that might be gained through better feature selection. Due to computation limitations, we only evaluated subsets of size ≤ 3 out of initial 50 generated features, and subsets of size ≤ 4 of initial 20 generated features.

Table 5.5 shows the results of these experiments. As expected, the performance of the resulting ESA queries was high, and at MAP of 0.3189 was even higher than the top keyword-based system we compared to (and far higher than the current optimal ESA-only result of 0.2223). Furthermore, fusing these results with BOW systems in Morag yielded far better results, with improvements in performance of up to almost 50%.

In addition, comparing the performance of the two experiments (best 4 out of 20 initial features and best 3 out of 50) shows that selecting out of a larger pool of features worked better despite fewer features being selected. This result indicates that if superior feature selection capability is available, it would be preferable to select from a longer prefix of the ESA concept vector. Note that a result for subsets of size ≤ 4 out of 20 features is an upper bound for subsets of size ≤ 3 out of 20, and thus this finding is valid even though the subset sizes were not identical. For some queries the optimal subset indeed was a smaller set, occasionally even a single feature, which indicates that using a *uniform* selection level parameter (θ) is not

BOW system	*Baseline*	MORAG(*optimal*4/20)) (MAP=0.2947)	MORAG(*optimal*3/50) (MAP=0.3189)
Xapian	0.2481	0.3322(+33.9%)	0.3692(+48.8%)
Okapi	0.2787	0.3406(+22.2%)	0.3714(+33.3%)
AT&T	0.2853	0.3475(+21.8%)	0.3673(+28.7%)
PIRCS	0.3063	0.3568(+16.5%)	0.3792(+23.8%)

Table 5.5: TREC-8 results with several BOW baselines, using optimal ("oracle") concept subset selection. *All* results were statistically significant.

an optimal strategy. Future work may investigate methods that utilize a per-query selection level, possibly using ideas such as *query clarity* [9].

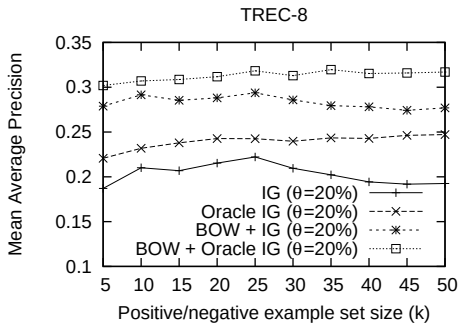

Figure 5.5: Concept-based performance (IG FS) using pseudo-relevant examples versus true relevant examples

Chapter 6

Related Work

Early approaches to concept-based IR attempted to leverage pre-existing conceptual thesauri such as Wordnet [35] for concept representation. Wordnet's synsets, like ESA concepts, represent real-life semantic human concepts and provide an intuitive, natural representation. Unlike ESA, the mapping method was not automatically generated by leveraging an existing resource, but rather by manual assignment of terms to synsets by Wordnet's editors. For example, in the Estonia-related example query, Wordnet's editors provided the synonymous "Esthonia" form for "Estonia," and "economic system" or "thriftiness" equivalents for the different meanings of "economy." Using such synonyms may assist recall to some limited extent, but it is clear that the "thriftiness" sense is not the intended one for the query, and using it would cause the retrieval to drift away, degrading system performance. Previous research has indeed shown the inconsistent improvement with this approach [52], which is successful only when applied manually [20] or augmented by other sources [34].

A major drawback of manually mapping words to concepts is the great effort invested in achieving good coverage of the domain language. Some researchers chose to overcome this obstacle by turning to *automatic* construction of a thesaurus from the target corpus itself, somewhat similar to the automatic construction of an ESA model from an *external* knowledge base (e.g., Wikipedia). [40] described a method for extracting a similarity thesaurus based on co-occurrence in the target corpus, thus obtaining more relevant concepts based on implicit domain knowledge, and yielding effective improvement. Another variant method combining the two approaches was suggested in [60], where a predefined dictionary of concepts was augmented with similar terms co-occurring in the corpus. Creating such co-occurrence resources is a computationally expensive process for large corpora, and one that needs to be constantly repeated for very dynamic corpora (such as the Web). With ESA-based concept representation the case is different, as the ESA

feature generator is built once, regardless of the actual corpus used and of corpus changes.

Another automated approach used document *ontologies* as a source for concept representation. One example, KeyConcept [19], is a retrieval system that maps documents to a limited subset of the concepts represented in the Open Directory Project[1] (ODP), using documents categorized to those concepts as training data for concept classifiers, and conducting search on the augmented text/concept representation. The use of ODP as a source for concept representation and automatic mapping has some parallels with our ESA approach, in particular when considering that ESA was implemented over ODP data as well [16]. However, the use of a limited concept ontology in KeyConcept resulted in a classifier that was not powerful enough to accurately classify the (short) queries. Thus query concepts were not automatically generated (as in this research) but had to be manually assigned by KeyConcept users. [6] describe another ontology-based approach, one that makes use of more formal semantic structures and queries, and combines semantic search with keyword-based retrieval to compensate for the knowledge base incompleteness. As with KeyConcept, this paper also assumes that semantic queries are created by the system user. The system was not evaluated on common IR benchmarks or against state-of-the-art IR systems.

Representing texts using concepts that are words, or explicit syntactic/semantic classes (such as Wordnet's synsets or ODP nodes), has the benefit of producing concepts that are human-readable, easy to analyze and reason about, and can be displayed to a user of such a system. ESA concepts, too, are based on human-defined natural concepts, as the example concept names throughout this paper show. Yet concepts may also be defined using latent semantics, with possibly broader concept coverage. By analyzing the latent relationships between terms in the target corpus, methods such as Latent Semantic Indexing (LSI) [10] can project the term space to a reduced-dimensions concept space, shared by documents and queries, and thus be applied successfully to the IR task [11, 25]. Like generating an ESA model or a co-occurrence thesaurus, generating an LSI model for a large corpus involves heavy computation. Unlike ESA, though, the generated LSI model is corpus-dependent, hence requiring the generation process to be repeated when the corpus changes or when a different corpus is used. In addition, the non-explicit nature of resulting concepts makes LSI difficult to tune and reason about [11]. More recent dimensionality reduction methods applied to IR have included Topic Models approaches [59] such as Latent Dirichlet Allocation [56] and the Pachinko Allocation Model [31].

All previously mentioned methods, including the one described in this paper, apply concept-based analysis to both the indexing and the retrieval stages of IR.

[1] http://dmoz.org

There also exists a large body of research applied to using concepts and ontologies in the retrieval stage only. Concept-based *query expansion* methods have been implemented using corpus-based similarity thesauri [40], domain-specific knowledge sources [33], or an ontology derived from Web sources such as Wikipedia [36]. But methods based on query expansion, in addition to the aforementioned representation-related issues, are also vulnerable to expansion-specific problems such as query drift and sensitivity to parameter tuning [4].

Chapter 7

Conclusion

We have presented a novel approach to concept-based IR using ESA as a representation method, introducing a feature selection component that is based on pseudo-relevance feedback. We have evaluated the proposed algorithms experimentally and demonstrated their improved performance. We have also estimated the potential for further improving the results of this approach, and outlined several insights in this regard that can guide future work.

Concept-based IR using ESA makes use of concepts that encompass human world knowledge, encoded into resources such as Wikipedia (from which an ESA model is generated), and that allow intuitive reasoning and analysis. Feature selection is applied to the query concepts to optimize the representation and remove noise and ambiguity. The results obtained by our proposed system (Morag) are significantly better than the baselines used, including those of top performing systems in TREC-8. Analysis of the results shows that improving the performance of the FS component is possible and will directly lead to even better results. In future work we plan to optimize the documents' representation as well, by leveraging recent work on compact ESA representations [32].

We believe the results we have shown in Section 5.3, coupled with the potential improvement demonstrated there, position ESA and the Morag framework as promising steps on the road to semantic retrieval solutions. Our work may provide both a leap in retrieval relevance and a potential shift in the IR paradigm, to one that is capable of manipulating human concepts rather than keywords only.

Bibliography

[1] James Allan, Jamie Callan, Fang-Fang Feng, and Daniella Malin. Inquery and trec-8. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 637–644, Gaitherburg, MD, 1999. NIST.

[2] Avi Arampatzis and Jaap Kamps. A study of query length. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 811–812, Singapore, 2008. ACM.

[3] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements that don't add up: ad-hoc retrieval results since 1998. In *Proceeding of the 18th ACM conference on Information and knowledge management (CIKM '09)*, pages 601–610, Hong Kong, China, 2009. ACM.

[4] Bodo Billerbeck and Justin Zobel. Questioning query expansion: an examination of behaviour and parameters. In *Proceedings of the 15th Australasian Database Conference*, pages 69–76, Darlinghurst, Australia, 2004. Australian Computer Society, Inc.

[5] James P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310, Dublin, Ireland, 1994. ACM/Springer.

[6] Pablo Castells, Miriam Fernandez, and David Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):261–272, 2007.

[7] Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: dataless classification. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 830–835, Chicago, IL, 2008. AAAI Press.

[8] Bruce W. Croft. *Combining Approaches to Information Retrieval*, chapter 1, pages 1–36. Kluwer Academic Publishers, 2000.

[9] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 299–306, Tampere, Finland, 2002. ACM.

[10] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[11] Susan T. Dumais. Latent semantic indexing (lsi) and trec-2. In *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 105–116, Gaitherburg, MD, 1994. NIST.

[12] Ofer Egozi, Evgeniy Gabrilovich, and Shaul Markovitch. Concept-based feature generation and selection for information retrieval. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1132–1137, Chicago, IL, 2008. AAAI Press.

[13] Edward A. Fox and Joseph A. Shaw. Combination of multiple searches. In *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 243–252, Gaitherburg, MD, 1994. NIST.

[14] Michael Fuller, Marcin Kaszkiel, Sam Kimberley, Corinna Ng, Ross Wilkinson, Mingfang Wu, and Justin Zobel. The rmit/csiro ad hoc, q&a, web, interactive, and speech experiments at trec 8. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 549–564, Gaitherburg, MD, 1999. NIST.

[15] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S.T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987.

[16] Evgeniy Gabrilovich and Shaul Markovitch. Feature generation for text categorization using world knowledge. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 1048–1053, Edinburgh, Scotland, 2005. Morgan Kaufmann Publishers Inc.

[17] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, pages 1301–1306, Boston, MA, 2006. AAAI Press.

[18] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1606–1611, Hyderabad, India, 2007. Morgan Kaufmann Publishers Inc.

[19] Susan Gauch, Juan M. Madrid, Subhash Induri, Devanand Ravindran, and Sriram Chadlavada. Keyconcept: a conceptual search engine. Tech. Report TR-8646-37, University of Kansas, 2003.

[20] Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan Cigarrin. Indexing with wordnet synsets can improve text retrieval. In *COLING/ACL Workshop on Usage of WordNet for NLP*, Montreal, Canada, 1998.

[21] Rakesh Gupta and Lev-Arie Ratinov. Text categorization with knowledge transfer from heterogeneous data sources. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 842–847, Chicago, IL, 2008. AAAI Press.

[22] Iryna Gurevych, Christof Muller, and Torsten Zesch. What to be? - electronic career guidance based on semantic relatedness. In *Association for Computational Linguistics (ACL)*, pages 1032–1039. The Association for Computer Linguistics, 2007.

[23] Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[24] David Hawking. Acsys trec-8 experiments. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 307–316, Gaitherburg, MD, 1999. NIST.

[25] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*, pages 50–57, Berkeley, California, 1999. ACM.

[26] Xiangji Huang, Yan Rui Huang, Miao Wen, Aijun An, Yang Liu, and Josiah Poon. Applying data mining to pseudo-relevance feedback for high performance text retrieval. In *Proceedings of the Sixth IEEE International Conference on Data Mining (ICDM'06)*, pages 295–306, Hong Kong, 2006. IEEE Computer Society.

[27] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning*, pages 121–129, New Brunswick, NJ, 1994.

[28] Marcin Kaszkiel and Justin Zobel. Effective ranking with arbitrary passages. *Journal of the American Society for Information Science and Technology*, 52(4):344–364, 2001.

[29] K. L. Kwok, L. Grunfeld, and M. Chan. Trec-8 ad-hoc, query and filtering track experiments using pircs. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 217–228, Gaitherburg, MD, 1999. NIST.

[30] Joon Ho Lee. Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 180–188, Seattle, WA, 1995. ACM.

[31] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pages 577–584, Pittsburgh, Pennsylvania, 2006. ACM.

[32] Sonya Liberman and Shaul Markovitch. Compact hierarchical explicit semantic representation. In *Proceedings of the IJCAI 2009 Workshop on User-Contributed Knowledge and Artificial Intelligence: An Evolving Synergy (WikiAI09)*, Pasadena, CA, 2009. Morgan Kaufmann Publishers Inc.

[33] Zhenyu Liu and Wesley W. Chu. Knowledge-based query expansion to support scenario-specific retrieval of medical free text. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, pages 1076–1083, Santa Fe, New Mexico, 2005. ACM.

[34] Rila Mandala, Tokunaga Takenobu, and Tanaka Hozumi. The use of wordnet in information retrieval. In *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*, pages 31–37, Montreal, Canada, 1998.

[35] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[36] David N. Milne, Ian H. Witten, and David M. Nichols. A knowledge-based search engine powered by wikipedia. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management*, pages 445–454, Lisbon, Portugal, 2007. ACM.

[37] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214, Melbourne, Australia, 1998. ACM.

[38] Rifat Ozcan and Y. Alp Aslandogan. Concept-based information access. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, pages 794–799, Las Vegas, NV, 2005. IEEE Computer Society.

[39] Martin Potthast, Benno Stein, and Maik Anderka. A wikipedia-based multilingual retrieval model. In *Proceedings of the 30th European Conference on IR Research (ECIR)*, pages 522–530, Glasgow, UK, 2008. Springer.

[40] Yonggang Qiu and Hans Peter Frei. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 160–169, Pittsburgh, PA, 1993. ACM Press.

[41] J Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[42] Stephen E. Robertson and Steve Walker. Okapi/keenbow at trec-8. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 151–162, Gaitherburg, MD, 1999. NIST.

[43] Joseph John Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, Englewood Cliffs, NJ, 1971.

[44] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[45] Hinrich Schuetze and Jan O. Pedersen. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, NV, 1995.

[46] Amit Singhal, Steve Abney, Michiel Bacchiani, Michael Collins, Donald Hindle, and Fernando Pereira. At&t at trec-8. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 317–330, Gaitherburg, MD, 1999. NIST.

[47] Amit Singhal, Mandar Mitra, and Chris Buckley. Learning routing queries in a query zone. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 25–32, Philadelphia, Pennsylvania, 1997. ACM.

[48] Amit Singhal, Gerard Salton, Mandar Mitra, and Chris Buckley. Document length normalization. Tech. Report TR95-1529, Cornell University, Ithaca, NY, 1995.

[49] Philipp Sorg and Philipp Cimiano. Cross-lingual information retrieval with explicit semantic analysis. In *Working Notes for the CLEF 2008 Workshop*, 2008.

[50] Christopher C. Vogt and Garrison W. Cottrell. Fusion via a linear combination of scores. *Information Retrieval*, 1(3):151–173, 1999.

[51] Ellen M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 171–180, Pittsburgh, PA, 1993. ACM.

[52] Ellen M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61–69, Dublin, Ireland, 1994. Springer-Verlag New York, Inc.

[53] Ellen M. Voorhees. Overview of the TREC 2004 robust retrieval track. In *Proceedings of the 13th Text REtrieval Conference (TREC-13)*, pages 70–79, Gaitherburg, MD, 2005. NIST.

[54] Ellen M. Voorhees and Donna Harman. Overview of the seventh text retrieval conference (trec-7). In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 1–24, Gaitherburg, MD, 1998. NIST.

[55] Ellen M. Voorhees and Donna Harman. Overview of the eighth text retrieval conference (trec-8). In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 1–24, Gaitherburg, MD, 1999. NIST.

[56] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*, pages 178–185, Seattle, Washington, 2006. ACM.

[57] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112, 2000.

[58] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 412–420, Nashville, Tennessee, 1997. Morgan Kaufmann.

[59] Xing Yi and James Allan. A comparative study of utilizing topic models for information retrieval. In *Proceedings of the 31st European Conference on IR Research (ECIR)*, pages 29–41, Toulouse, France, 2009. Springer.

[60] Xiaohua Zhou, Xiaodan Zhang, and Xiaohua Hu. Using concept-based indexing to improve language modeling approach to genomic ir. In *Lecture Notes in Computer Science*, pages 444–455, London, UK, 2006. Springer.

[61] Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, Melbourne, Australia, 1998. ACM Press.

מושגים. לבסוף, שיפרנו עוד את ביצועי המערכת על-ידי שימוש בטכניקת מיזוג מידע (Data Fusion), כאשר אנו ממזגים את תוצאות האחזור מבוסס-המושגים עם תוצאות אחזור מבוסס-מילים רגיל אשר שימש אותנו גם ליצירת הדוגמאות לתהליך הבחירה.

ביצענו בחינה מעמיקה של ביצועי המערכת שבנינו ("מורג") כדי לבדוק את ביצועי המערכת ולחקור כיצד ניתן יהיה לשפר את ביצועיה אף יותר בעתיד. הביצועים נבדקו על מבחנים מתוך ועידת TREC, אשר משמשת כאבן הבוחן המקובלת ביותר למערכות אחזור מידע ומול מערכות אשר השיגו את התוצאות הטובות ביותר במבחנים אלו. התוצאות הראו ביצועים העולים באופן משמעותי על אלו של רוב המערכות שנבדקו, וזאת על אף שהראנו שיש יסוד להניח שהביצועים מוערכים בהערכת חסר עקב הטייה מובנית במתודולוגית TREC כנגד מערכות מסוגה של מורג.

מורג בנויה כמערכת מודולרית, ומאפשרת להשתמש בכל מערכת אחזור מבוסס-מילים קיימת בתור מערכת הבסיס. השתמשנו בעובדה זו כדי לבצע סדרת ניסויים שבה הראנו שיפור בכל אחת מהמערכות המובילות במבחן הנ"ל, על אף שהיו הטובות מסוגן. בנוסף, בדקנו איזה חלק מהשיפור נזקף לזכות מיזוג המידע כשלעצמו, ולצורך כך ביצענו מיזוג של כל מערכת אחזור מבוסס-מילים שנבדקה עם מערכת אחזור מבוסס-מילים אחרת שרמת ביצועיה דומה לזו של האחזור מבוסס-המושגים לבדו. השיפור שהושג היה קטן משמעותית מזה שהושג במורג, דבר שהצביע על הערך המוסף המשמעותי של אחזור מבוסס-מושגים.

לבסוף, בדקנו בסדרת ניסויים נוספת מה השיפור הפוטנציאלי שעוד צפוי ע"י שיפורים בתהליך בחירת התכוניות. התוצאות הראו מקום רב לשיפור נוסף, שניתן להשגה מתוך שיפור קבוצת הדוגמאות שנבחרת לשימוש בתהליך הבחירה, ומשיפור אלגוריתם הבחירה עצמו.

לסיכום, השימוש באחזור מבוסס-מושגים בעזרת ניתוח סמנטי מפורש, כפי שהודגם ע"י מערכת מורג במחקר זה, מייצג קפיצה ביכולות האחזור יחסית למערכות קיימות מבוססות-מילים. המערכת שהוצגה בעלת ארכיטקטורה פשוטה ומודולרית, ומשיגה תוצאות העולות על שיטות שנחשבו עד כה לטובות ביותר בתחום. המחקר אף הראה אינדיקציות לשיפור פוטנציאלי משמעותי נוסף אשר יכול להיות מושג ע"י מחקרי המשך.

במחקר זה אנו מציגים שיטת אחזור מבוסס-מושגים חדשה המבוססת על ניתוח סמנטי מפורש (ESA). ESA הינה שיטת יצוג טקסט שהוצעה לאחרונה היכולה להרחיב יצוג מבוסס-מילים ע"י תכוניות מבוססות-מושגים, תכוניות אשר מיוצרות אוטומטית מתוך מאגרי ידע אנושי מקיפים כגון ויקיפדיה. המושגים ושיטת הייצוג של טקסט כלשהו בהם נבנים בצורה אוטומטית מתוך מאגר הידע החיצוני. בזכות ההסתמכות על מקורות מידע בעלי מבנה כמו ויקיפדיה, המושגים שבהם עושה הייצוג שימוש הינם מושגים המייצגים מושגים אנושיים אינטואיטיביים. לבסוף, בזכות הכיסוי הרחב של מאגרי ידע כמו ויקיפדיה, בניית עולם המושגים ופונקציית המיפוי אליו יכולה להתבצע באופן חד-פעמי ולהתאים לשימוש על קורפוסי מטרה כלשהם.

במחקרנו מצאנו ששימוש פשטני בתכוניות המיוצרות ממסמכים ומשאילתות על-מנת לבצע אחזור במרחב זה אינה מביאה מיד לתוצאות המצופות, וזאת מכיון שהייצוג הבסיסי מכיל רעש ואי-דיוקים, הנובעים משיטת הבנייה האוטומטית של הייצוג ומאופי המידע בויקיפדיה. כדי להתגבר על בעיות אלה, השתמשנו בשיטות של בחירת תכוניות (Feature Selection), על מנת לבחור את סט התכוניות בייצוג השאילתא שיביא לתוצאות מיטביות.

כאשר מחקר קודם השתמש בשיטת ESA כדי לשפר את ביצועי משימת קיטלוג טקסטים, שלב בחירת התכוניות היווה חלק חשוב מהתהליך וסינן חלק גדול מהרעש, ואכן תוצאות השימוש ב-ESA שם הראו שיפור מרשים בביצועים. אולם, משימת אחזור מידע אינה משימה מסוג של למידה מונחית ולכן לא זמינות בה דוגמאות מתוייגות כפי שקיימות במשימת קיטלוג טקסטים. לכן היה עלינו למצוא מקור מידע אחר שישמש להערכת תרומתה של כל תכונית לתוצאות האחזור.

נבחין כי קיימות מערכות אחזור מידע אשר דומות במהותן ללמידה מונחית. מערכות המשלבות משוב מהמשתמש מאפשרות למשתמש לתייג מסמכים שאוחזרו ע"י המערכת כרלוונטיים או לא-רלוונטיים, ובכך לאפשר למערכת לבנות שאילתא מוצלחת יותר ולהציג את תוצאותיה למשתמש. מחקרים הראו שניתן להשיג שיפור בביצועי המערכת גם ללא משוב ישיר מהמשתמש, ע"י התייחסות למסמכים הראשונים בתוצאות האחזור כמסמכים רלוונטיים, ושימוש בתוכנם כדי לשפר את השאילתא הראשונית. גישה זו נקראת Pseudo-Relevance Feedback, ועל בסיסה יצרנו דוגמאות מתוייגות עבור תהליך הבחירה.

לשם כך הפכנו את האחזור לדו-שלבי - בשלב ראשון מבצעת המערכת אחזור מבוסס מילות מפתח סטנדרטי, ותוצאות אחזור זה משמשות לבניית סט הדוגמאות הפסבדו-רלוונטיות: המסמכים שדורגו ראשונים מתוייגים כמסמכים רלוונטיים, ואלו שדורגו אחרונים מתוייגים כלא-רלוונטיים. כעת אנו מפעילים מספר שיטות בחירה אשר מחשבות סטטיסטיקה של מידע אשר כל תכונית מסוגלת לתרום בהפרדה בין המסמכים הרלוונטיים ואלה שאינם רלוונטיים, ובכך להעריך את מידת התאמתה לשימוש בשאילתא הסופית.

בנוסף, מצאנו שפירוק המסמכים הממופתחים לפסקאות שכל אחת מהן מיוצגת ומאוחזרת בנפרד תורם משמעותית לשיפור הדיוק באחזור. שימוש בפסקאות קצרות בתור דוגמאות חיוביות ושליליות עדיף על שימוש במסמכים מלאים מכיון שדוגמאות אלה הינן ממוקדות יותר במושגים המיוצגים בהם, לעומת מסמכים שלמים אשר עלולים להכיל מספר רב של

תקציר המחקר

מערכות אחזור מידע עושות לרוב שימוש במילות מפתח כדי למפתח ולאחזר מסמכים. אחזור המבוסס על מילות מפתח עלול להחזיר תוצאות שאינן שלמות או שאינן מדויקות כאשר אותו מושג אנושי מתואר במלים שונות במסמכים ובשאילתות. לדוגמא, כותב מסמך מסוים עשוי להשתמש במילה "מסך" בתאור חלקי מערכת מחשב, כאשר משתמש במערכת האחזור עשוי לבנות שאילתא המשתמשת במילה "צג". בנוסף, מילים עלולות להיות רב-משמעיות ולגרום לאחזור מסמכים שאינם בנושא הרצוי. כך למשל המילה "מסך", כאשר תתוסף לשאילתא, תביא להחזרת תוצאות המתייחסות גם למסך על בימת תאטרון. יתרה מזאת, היחס בין אותן מילים שונות עשוי לעתים לחרוג מיחס מילוני פשוט של מילים נרדפות ולהתבסס על קרבה סמנטית. כך למשל מסמך העוסק בהתקני תצוגה למחשב עשוי להיות רלוונטי לשאילתא הנ"ל, גם אם המילה "מסך" לא הוזכרה בו במפורש. הבחנה ביחס כזה עלולה לדרוש נגישות למאגרים נרחבים של ידע עולם אנושי, שכן הידע אשר קושר מילים אלה למושג אחד עשוי לחרוג מהידע המצומצם הקיים במילונים ובאגרונים.

שיטות מבוססות-מושגים לאחזור מידע ניסו בעבר להתמודד עם בעיות אלו תוך שימוש במקורות מידע שונים. שימוש באגרונים שנבנו בעבודה ממושכת של עורכים יכול לסייע במציאת מילים נרדפות ואף הכללות, אולם מורכבות העבודה הנדרשת לבניית מילון שיקיף את כל המידע הנדרש גורמת לכך שגישה זו אינה ישימה לקנה מידה גדול. גישה אחרת ניצלה מידע אודות צמידות בהופעת מילים בקרבה בקורפוס רחב של מסמכים. מידע כזה יכול לרוב להצביע על קרבה בין מילים גם כאשר קרבה זו אינה בהכרח מילונית. יתרונה של גישה כזו על הקודמת הוא גם בהיותה אוטומטית, אם כי זמן החישוב למידע כזו הינו רב, ויש לחזור עליו עם השתנות המסמכים בקורפוס. גישה אחרת ניסתה לחלץ מושגים חבויים בטקסט תוך שימוש בניתוח סמנטי חבוי (לטנטי), אשר מאפשרת מציאת קשרים עמוקים בין מילים שונות על סמך המידע המיוצג במסמכי הקורפוס, אולם שיטה זו היא אף יותר עתירת חישובים. בנוסף, המושגים המתקבלים בה הינם מתמטיים ביסודם ואינם מייצגים ישירות מושגים אנושיים.

המחקר נעשה בהנחיית פרופ' שאול מרקוביץ' בפקולטה למדעי המחשב.

אחזור מידע מבוסס-מושגים תוך שימוש בניתוח סמנטי מפורש

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
מגיסטר למדעים במדעי המחשב

עופר אגוזי

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
חשוון ה'תש"ע חיפה נובמבר 2009

אחזור מידע מבוסס-מושגים תוך שימוש בניתוח סמנטי מפורש

עופר אגוזי