VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui

School of Engineering and Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

Improving Automatic Query Expansion

Glen Robertson

Supervisor: Dr Xiaoying Gao

October 16, 2009

Submitted in partial fulfilment of the requirements for
Bachelor of Information Technology.

Abstract

Current web search engines often produce unsatisfactory result sets, where the top ranking documents represent only a subset of the aspects given in the user's search query. The aspects in a search query that are not represented well in the result set are called underrepresented aspects. An Automatic Query Expansion algorithm called AbraQ, was recently developed at Victoria University. AbraQ identifies the underrepresented aspects in a user's search query, identifies keywords to better represent those aspects, and adds these keywords to the original search query in order to return a better result set.

This project aims to improve AbraQ so it produces better quality expanded queries. The following areas of improvement are investigated. 1: New methods are investigated to build vocabularies for each aspect in the query. 2: A new component is added to classify the relevance of individual documents in the result set. 3: New sources for expansion terms are explored, to find whether there are better sources than those used in AbraQ. The results from a series of experiments show that the new methods help AbraQ to produce better quality expanded search queries.

Contents

Chapter 1

Introduction

Today, the Internet consists of approximately 28 billion documents [1]. Because of this large quantity of documents, it is important that users have the ability to find the documents that they need efficiently. Users find the documents that they need through the use of search engines, by entering a search query, consisting of 2-4 words on average to express their search goal [2]. A user's search query often has multiple aspects. An aspect is a group of consecutive words in a query that describe a certain concept. For example, the search query "Black bear attacks" has two aspects: "Black bear" and "attacks". A relevant document is one that contains information about all of the aspects in the user's query. It is often the case that the top-ranked documents only contain information about some of the aspects of the users query, and the remaining aspects are only covered briefly. These aspects are called underrepresented aspects. Aspects may be underrepresented in a result set because of word mismatch. This is where the words that a user uses to describe an aspect are different to the words used by the authors [3]. A possible way to solve the word mismatch problem is to append words to the query that are similar to the underrepresented aspects. This task is called query refinement and is often performed manually by users. This is time consuming because they have to look at the retrieved pages to work out how to modify their query to filter out the irrelevant results [4]. A useful idea is to refine the query automatically, without any additional input from the user after entering their search query. This refinement process is called Automatic Query Expansion.

1.1 Automatic Query Expansion

Automatic Query Expansion (AQE) is a process, which reformulates the user's original search query, by automatically modifying existing terms or adding new terms to the search query. One problem with query expansion is query drift, which is the change in the underlying "intent" between the original query and it's expanded form [5]. One method of preventing query drift is to measure the relevance of the documents from the expanded query against the original query, to ensure that the documents are closely matched to the original search goal.

1.2 Goals and motivations

An AQE algorithm called AbraQ was recently developed by Daniel Crabtree at Victoria University in 2007. AbraQ is described in [4], where it showed good quality improvements for hard search queries, compared to previous query expansion techniques. However, AbraQ still needs much work before it is reliable.

The main goal of this project is to improve AbraQ, so it is developed to a reliable state, where the queries produced will always improve the quality of the result set and never make it worse. If this can be achieved, it can be safely integrated with existing search engines. This project will identify areas where AbraQ needs improvement, and a range of methods will be investigated in attempt to improve these areas. The areas that will be investigated are described below.

1.2.1 Aspect vocabularies

An aspect is a phrase of words from the user's query that describes a particular concept. AbraQ builds vocabularies for each aspect in the search query, where each vocabulary consists of a set of phrases that are semantically related to the aspect. These phrases are ordered using some ranking method where higher ranking phrases have a stronger semantic relation to the aspect. The phrases in the vocabularies generated by AbraQ appear to have a weak relationship with their aspect. Each phrase in an aspect's vocabulary should have a strong semantic relationship because these phrases are used to identify underrepresented aspects in the original document set, and they are also used to generate expanded queries. One limitation of AbraQ is that the vocabulary phrases are currently limited to one word per phrase. It would be useful to add phrases rather than single words. For example, if we are building a vocabulary for the aspect "America", it is only useful to add "United States" as a vocabulary term, rather than "United" or "States" by itself. The method of retrieving terms and the ranking method will also be investigated.

1.2.2 Query-Document relevance classification

AbraQ does not have the ability to identify the documents that are relevant to the user's search query. It is useful to add this functionality so we can identify whether or not an expanded query helps to improve the quality of the resulting document set that is returned to the user. In addition to this, the relevant documents are useful because they will contain good quality expansion phrases.

1.2.3 Expansion term sources

The most important part of the query expansion process is to find good quality terms to append to the user's search query. AbraQ only uses the vocabulary to find expansion phrases, which are often poor quality. Such phrases can hurt the precision of the query results if they are not related to the original query. Additional sources for expansion terms will be explored to try and produce better quality expanded queries.

1.3 Report overview

Chapter 2 contains background and previous work that relates to AQE, followed by an in-depth description of the AbraQ algorithm in Chapter 3. Chapters 4, 5 and 6 will focus on each component involved in AbraQ, identifying and attempting to implement areas of improvement for each respective component, followed by a series of experiments that test whether or not the proposed improvements are effective. The effective improvements will be implemented in the final algorithm. Chapter 6 will also compare the performance of AbraQNew to AbraQ and existing query expansion algorithms. Finally, Chapter 7 will summarize the contributions that have been made, mention any limitations of the final algorithm, and discuss ideas for future work.

Chapter 2

Background

This chapter introduces previous work related to Automatic Query Expansion.

2.1 History

Previous query expansion research has explored the use of different techniques to find the best expansion terms. These techniques are described below.

2.1.1 Relevance Feedback

Relevance Feedback is a technique, which expands queries using keywords extracted from relevant documents in the user's search query. Early attempts were manual, in the sense that the user had to select the relevant documents [6], or relevant clusters [7]. Later attempts introduced a technique called Pseudorelevance feedback. This is where the algorithm assumed that the set of top-ranked documents were relevant, so less user interaction was required. This technique has become widely used in recent years, however, its performance is known to be very erratic, where retrieval can be hurt in the case that most of the top-ranked documents are not relevant [8].

2.1.2 Co-occurrence

Co-occurrence based techniques expand queries using keywords that co-occur frequently with any query terms. One paper that explores co-occurrence for query expansion is "A study of the effect of term proximity on query expansion" [9]. In this paper, a range of distance functions are compared, arguing that the semantic relatedness between terms weakens with the increase in distance separating them. This is described in more detail in 2.2.1.

2.1.3 Thesaurus-based techniques

One technique for finding new terms is to use a Thesaurus or WordNet. These provide terms which have a similar meaning to a given word in the query. The disadvantage of this approach is that each word only relates to one word in the query, where techniques such as relevance feedback have the advantage of finding words that may relate to all the words in the query. WordNet is a lexical database for the English language. It groups English words into sets of synonyms called Synsets, which provide short, general definitions, and record the various semantic relations between these synonym sets [10]. There appears to be little evidence that thesaurus-based techniques are useful expansion terms. [11] shows no major improvements when attempting to use a WordNet database to expand queries.

2.2 Related Work

This section describes ideas from past research, related to query expansion.

2.2.1 A study of the effect of term proximity on query expansion

[9] explores a range of distance functions to measure term proximity. The idea is that useful expansion terms can be gained from the top-ranked documents of the original query, where the most closely semantically related terms occur in close proximity to the query terms. When building a set of terms for a query, a common approach for term selection is to use a distance factor, where terms are weighted based on their word distance from the query term. One argument is that the strength of association between words decays exponentially with the increase in distance. This paper experiments with several distance factors, and combines them with a mutual information measure.

Mutual information

The mutual information (MI) score between a pair of words compares the probability that the two words occur as a joint event with the probability that they occur individually and that their co-occurrences are simply a result of chance [9].

The standard formula for calculating mutual information is:

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)} \tag{2.1}$$

$P(x,y)$: the probability that x and y occur together.
$P(x), P(y)$: the probabilities that x and y occur individually.

The important conclusion to draw from this paper is that the combination of collocation distance, collocation frequency and mutual information helps to select better query expansion terms than the use of MI or distance functions alone.

2.2.2 The Google similarity distance

[12] introduces a mutual-information based technique that measures the semantic relationship between two phrases by leveraging the large size of Google's document corpus. It is an efficient way to reinforce the semantic relationship and is shown to improve with the size of the corpus. The technique analyses the search counts of query results, where two phrases are compared by querying them separately and then as a combined phrase. The idea is to measure how commonly the two phrases occur together in documents compared to occurring separately in other documents. The Normalised Google Distance is a version of this formula that returns a number approximately between 0 and 1. The formula is as follows:

$$NGD(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log(N) - \min\{\log f(x), \log f(y)\}} \tag{2.2}$$

$f(x)$: the number of pages containing x.
$f(y)$: the number of pages containing y.
$f(x,y)$: the number of numbers containing both x and y.
N: the total number of documents in the Google corpus.

The formula above may be useful when finding good expansion terms for a query. An idea

is to use the formula as a method to rank an aspect's vocabulary. This would ensure that the top terms have a strong semantic relation with the aspect, making the top terms better candidates for expansion terms.

2.2.3 Exploiting underrepresented aspects for automatic query expansion

[4] introduces the AbraQ algorithm. The algorithm focusses on improving hard queries, that is, queries that do not return many relevant results. It identifies the aspects in a query, then identifies which aspects are underrepresented in the result set. It then appends a term to the query that is closely related to the underrepresented aspect, in order to increase the chances of finding documents that better represent that aspect and are therefore more relevant to the user.

Local Document Analysis

Local Document Analysis is a technique which analyses the content of a set of documents in the corpus. In AbraQ, this is used to build vocabularies for each aspect in the query, where each vocabulary consists of a set of terms that are closely related to the aspect. These vocabularies are used to determine which aspects in the original result set are underrepresented, and then finding useful terms to improve the underrepresented aspects.

Global Document Analysis

Global Document Analysis is a technique which analyses properties of a search engine's entire document corpus. AbraQ utilizes Global Document Analysis to identify the aspects in a query, and also assists the vocabulary building stage by re-ranking the terms discovered using Local Document Analysis.

2.3 The AbraQ algorithm

2.3.1 Overview

AbraQ is an AQE algorithm that claims to make significant improvements in web search performance. AbraQ performs three main steps:

1. Identify the aspects of the query.
2. Identify which aspects are underrepresented in the result set.
3. Identify refinement term(s) to address the under-represented aspects.

These steps are described in detail below.

2.3.2 Identify aspects

An aspect is a group of consecutive words in a query. Splitting the search query into aspects allows us to identify different topics of the query, where we can then identify which topics in the query require expansion. For example, given the query "black bear attacks", it would be detrimental to expand on the word "black" alone. Terms relating to the colour "black" would be added, but they will not help improve the query because the user is in fact searching for documents relating to "black bears". The process of grouping consecutive words into aspects uses search engine document counts.

Algorithm

Two semantic measures are used to determine whether a sequence of words is a valid aspect. For a given sequence of words, "Existence" is a measure of how frequently the sequence occurs relative to the set of words. "Support" is a measure of how frequently the sequence occurs relative to all other permutations of those words.

Given a sequence of words s, the existence and support can be calculated as follows:

$$Existence(s) = \frac{DP(s)}{D(s)} \tag{2.3}$$

$$Support(s) = \frac{DP(s)}{\sum_{s' \in Perm(s) \setminus \{s\}} DP(s')} \tag{2.4}$$

If $Existence(s) \cdot Support(s) \geq 1.0$, then s is considered a valid aspect.

DP(s) is the number of documents that contain s as a phrase. Using the Google search engine, this value can be retrieved by performing the query for s in double-quotes.

D(s) is the number of documents that contain each of the words in s, as long as each word occurs somewhere in the document. This value can be retrieved from Google by performing the query for s.

$\sum_{s' \in Perm(s) \setminus \{s\}}$ is the total number of documents that contain permutations of s, other than s itself. This can be obtained by finding the permutations of s. There are $(n! - 1)$ permutations excluding s itself, where n is the number of words in s. Then, we surround each permutation phrase in double quotes and concatenate them with the OR operator. Performing all permutations in one query has the advantage that it finds the intersection of all documents. Conversely, if we query each permutation as an isolated query, the accumulated number of documents would be higher because the documents that intersect the result set of each query would be counted multiple times.

2.3.3 Identify underrepresented aspects

Vocabulary

AbraQ constructs a vocabulary model for each aspect by running sub-queries for all aspects and all pairs of aspects which have been identified. It finds all the terms that occur in these documents, currently limited to the top 10 documents for each sub-query. It then ranks each term according to the document frequency: the amount of retrieved documents that contain the term from the sub-queries that contain the aspect. For each aspect, the top 200 terms with the highest document frequency are added to the vocabulary model, and these top 200 terms are filtered to the top 50 terms, by re-ranking by co-occurrence strength between each term and the aspect. The co-occurrence strength is a ratio of the actual and expected co-occurrence frequency of a term t and an aspect a. The actual co-occurrence frequency is the fraction of documents that contain both the aspect and the term. The expected co-occurrence frequency is the product of the fraction of documents that contain the aspect and the fraction of documents that contain the term.

Calculate document aspect scores

AbraQ scores the vocabulary model of each aspect a against the documents from the original query q. First a raw aspect score is calculated for each aspect, which is the dot product of the weights in the aspect's vocabulary model and the term frequencies in the documents from the original query. The raw aspect scores are then normalized so they sum to one, resulting

in a set of relative aspect scores (RAS). A particular aspect is underrepresented if it's RAS is under a set threshold. The threshold is calculated from the number of aspects as follows:

$$Threshold(q) = \frac{1}{|a| + 1} \tag{2.5}$$

$$RAS(a, q) < Threshold(q) \rightarrow a \text{ is underrepresented} \tag{2.6}$$

2.3.4 Identify refinement

AbraQ only attempts to expand the query if it identifies any underrepresented aspects. For the most underrepresented aspect, AbraQ generates a set of expanded queries. Each query consists of the original query plus a high weighted term from the aspect's vocabulary model. A set of relative aspect scores are calculated for each expanded query, by using the result set of the expanded query and the vocabulary models of the original set of aspects. A score is then calculated for each expanded query, which compares the set of raw aspect scores for the expanded query to the relative aspect scores of the original query.

$$RS(q') = \sum_{a} \frac{RAW(a, q')}{RAS(a, q)} \tag{2.7}$$

The query with the highest score is the query returned to the user.

Chapter 3

AbraQNew algorithm

The improved version of the AbraQ algorithm that is being developed throughout the report will be referred to as AbraQNew. This chapter will provide an overview of AbraQNew. The chapters that follow will describe the different components of AbraQ, and how they have been developed to produce AbraQNew.

In both AbraQ and AbraQNew, the input is the user's search query, and the output is the expanded query. AbraQNew shares the same overall structure with AbraQ, in that it performs the same steps: Identifying aspects, identifying underrepresented aspects, and identifying the refinement to produce the expanded query. However, AbraQNew uses different techniques for each step, so a higher quality expanded query is returned to the user.

3.1 Overview

Each task is described below in respect to how it will be changed in AbraQNew.

3.1.1 Identify aspects

The aspect identification component is for the most part, the same as the AbraQ algorithm. After some initial testing of the aspect identification component, the word groupings were for the most part, satisfactory. A satisfactory grouping of words was determined from my own intuition. Some typical aspects identified by the AbraQ algorithm are shown in Table 3.1.

Aspects	**Computed aspects**
black bear, attacks	black bear, attacks
hubble telescope, achievements	hubble telescope, achievements
thomas, the, tank engine	thomas, the, tank engine
computer science	computer science
artificial intelligence	artificial intelligence
george bush	george bush
bass guitar	bass guitar
united states of america	united states of america
helen clark, prime minister	**helen, clark**, prime minister
air new zealand	**air, new zealand**
victoria university	**victoria, university**

Table 3.1: Aspects identified by AbraQ. Incorrect identifications are in bold.

One observation that can be made from Table 3.1 is that some aspects are not classified correctly when they do not exist enough on the Internet. It groups words as aspects based on how often the phrase of words occurs in the global document set. Therefore, an aspect is valid if it is a popular topic on the Internet.

It is hard to find a better method of identifying aspects and there is little previous research related to aspect identification. For lack of a better method, this component of the algorithm is left unmodified.

3.1.2 Identify underrepresented aspects

AbraQNew will attempt to produce a better set of vocabulary terms. If a better vocabulary can be built, the accuracy of aspect scores will improve. This will improve the accuracy of the underrepresented aspect identification process. The following ideas will be investigated:

- Utilizing additional sources such as a WordNet.
- Increasing the number of documents analysed in sub-queries
- Applying different methods of weighting the terms found using sub-queries.
- Identifying phrases consisting of more than one word from documents in sub-queries. AbraQ can currently only identify single word terms.

3.1.3 Identify refinement

AbraQ uses the highest ranked terms from an underrepresented aspect's vocabulary for expansion terms. A better vocabulary will result in better quality expansion terms.

AbraQNew will utilize a range of different sources as well as the vocabulary. Expansion terms from the original query are good quality if they come from relevant documents. Therefore, it is also useful to develop a method of classifying relevant and irrelevant documents automatically. AbraQ only considers the quality of an expanded query by the sum of the aspect scores for each document. AbraQNew will determine the best expanded query by the number of relevant documents that it produces.

3.2 Implementation details

3.2.1 Overview

AbraQNew is written in the Java programming language. It is currently limited to using the Google search engine. A Google class has been constructed to provide methods for retrieving page counts and URLs from a search result set, when given a search query string as input. Pages from the result set are retrieved using a scraper that pretends to be a web browser. The Google class utilizes the Google JSON Search API [13] to retrieve page counts. It is important to use the API where possible because an unlimited quantity of requests can be performed on the Google search engine, where Google will block a traditional scraper after performing too many requests. To access the WordNet database, a prebuilt Java WordNet interface [14] is used. This interface requires a word or phrase and it's Part-of-Speech type as input to output a set of synonyms. Each query aspect is tagged using a Part-of-Speech tagger [15]. Different Part-Of-Speech types return different synonym sets for the same word. For example: "attacks" as a noun returns the set: {onslaught, onset, onrush} and "attacks" as a verb returns the set: {assail}.

3.2.2 Stemmed Phrases

A StemmedPhrase class has been constructed, which stores the stemmed version of a phrase for comparison and all unstemmed variations along with the number of times each unstemmed variation has occurred. For a particular StemmedPhrase object, the unstemmed variation that occurs the highest number of times is used for output. Two StemmedPhrase objects are compared using the stemmed version of the word, so "running" will be equal to "run". An implementation of the Porter stemmer is used [16]. A vocabulary is a sorted set of StemmedPhrase objects with a floating-point weight. The set is sorted first by weight, and StemmedPhrase objects of equal weight are sorted alphabetically. If a new phrase/weight pair is added, it's stemmed version is compared with all stemmed phrases in the existing vocabulary set. If there is a match, the weight is incremented by the new pair's weight and the unstemmed phrase is added to the collection of unstemmed variations for that existing pair.

Chapter 4

Aspect vocabularies

This chapter focusses on producing better vocabularies for aspects. AbraQ uses the vocabulary for identifying underrepresented aspects and identifying the refinement. The degree to which a particular query can be improved depends on the quality of the refinement, where the terms with the highest weighting will be refinement terms. If a good quality vocabulary is built and used for refinement, then it is more likely that the result of our search query can be improved, through identifying the correct underrepresented aspects and identifying good quality refinements. Conversely, when a poor quality vocabulary is used, it is likely that underrepresented aspects may be incorrectly identified, or poor expansion terms may be applied, resulting in no improvements to the query.

A vocabulary for an aspect is a set of term/weight pairs, ordered by the highest weighting first. A term consists of the original term's phrase and a stemmed version of that phrase. The stemmed version is used for comparison. Each time that a term is added that has an equivalent stemmed version existing in the vocabulary set, the weight of the existing term/weight pair is incremented by the new term's weight. If the original phrase differs from the existing phrase, then the new phrase is also remembered by the term/weight pair along with the amount of times that the phrase has occurred. This allows us to group different variations of the same phrase, so when we use the term for expansion, we can retrieve the most popular variation.

For example, in the aspect set: "attacks", the term/weight pair with the highest weight is the stemmed phrase: "kill people". The stemmed version of the term has been added to the vocabulary set three times. The first occurrence was the phrase: "kills people", the second: "killed people", the third: "kills people". The output phrase for this term/weight pair will be: "kills people", because it has occurred 2 times and "killed people" has only occurred once.

For a given aspect, this chapter explores two potentially useful sources for vocabulary terms:

- The set of top-ranked documents from any sub-query that includes that aspect
- The synonym set for the aspect from a WordNet database

4.1 Sub-query terms

In AbraQ, sub-queries are built from all aspects and aspect pairs. The top 10 documents are analysed from each sub-query, and the terms are ranked by document frequency.

These terms are added to the vocabularies for each aspect that is included in the sub-query, where terms from sub-queries with a pair of aspects are added to both aspects in the pair. The term's weighting is incremented in the vocabulary set by the document frequency, divided by the number of aspects in the sub-query.

4.1.1 Formulating sub-queries

AbraQNew will modify the sub-query set, where sub-queries are built from all possible combinations of aspects in a sequence from left-to-right, rather than just limiting to single aspect and aspect pair sub-queries.

This is a minor improvement that compensates for the possibility that aspects may be split too much by the aspect identification component. An aspect pair that is made of two non-consecutive aspects in the query may lose semantic information, especially if the aspects are split too much. For example, if the query "black bear attacks" was split into aspects as invididual words, then the sub-query "black attacks" would be constructed. Because AbraQNew only considers consecutive phrases when grouping into aspects, "black attacks" will not be a sub-query generated by the system.

Table 4.1 compares the set of sub-queries generated by AbraQ and AbraQNew for the query: "fatal black bear attacks in north america". This query has the following aspects: {**fatal, black bear, attacks, in, north america**}. "in" is ignored as an aspect because it is a common stop-word.

AbraQ set	**AbraQNew set**
fatal	fatal
black bear	fatal black bear
attacks	fatal black bear attacks
north america	black bear
fatal black bear	black bear attacks
fatal attacks	black bear attacks north america
fatal north america	attacks
black bear attacks	attacks north america
black bear north america	
attacks north america	

Table 4.1: Sub-queries generated by AbraQ and AbraQNew for "Fatal black bear attacks in North America"

When building the vocabularies, documents from sub-queries are disregarded when they intersect the set of documents from the original query. This is because any documents used to build vocabularies from the original query will have an unfair advantage in the relevance detection step. If this occurs, then these documents will be more likely to gain higher aspect scores. It is important to keep the documents from the original query independent from the documents used to build the vocabularies.

4.1.2 Distance functions

As mentioned above, AbraQ weights each term based on document frequency, which is where each term is weighted based on the amount of documents that they occur in out of the total amount of documents that also include the aspect. [9] shows promise in weighting terms using distance functions, combined with mutual information measures.

In AbraQNew, a number of different weighting methods are considered. The distance functions apply higher weightings to terms that occur closer to occurrences of an aspect within the document set. There are two differing methods of gathering terms from a document and ranking them using the distance functions. These methods are described for a document d, containing occurrences of the aspect a.

- Word window method: The method searches d for occurrences of a. For each occurrence, a collection of terms are retrieved before and after the occurrence, limited by a predefined word distance. For example, if the predefined word distance was five, we retrieve the five terms before the aspect occurrence, and five terms after the aspect occurrence. A weighting is calculated for each term based on the amount of words between the current term and the aspect occurrence. This term/weighting pair is then added to the vocabulary set. This method easily filters out what might potentially be irrelevant terms, by assuming that related terms are those which occur closely to aspect occurrences.

- Whole document method: The method searches d for occurrences of a and constructs a set of indexes, where each index represents the position of each aspect occurrence in d. The method then iterates through each term t in d (skipping the aspect occurrences). A weighting is calculated for t based on the distance between t and the closest aspect occurrence. This term/weighting pair is then added to the vocabulary set. It differs from the word window method in that all terms in the document are considered, and each term occurrence is only added once. One advantage of this method is that it gives terms that occur further away from aspect occurrences a chance to belong to that aspect's vocabulary. Seen as though the document results from a query that contains the aspect, there is still a chance that terms occurring further away may still be relevant. The disadvantage is that it is less efficient than the word window method, because more memory is required to store a larger set of terms, and more computation is required because more terms have to be sorted to gain the higher ranked terms. Because the focus of this study is to improve the quality of the expanded queries, it is more important to use the whole document method, and favour quality in the tradeoff with efficiency.

There are three distance functions that can be used to rank terms based on word distance:

- Linear distance (linDist):

$$linDist(t,a) = \frac{1 + w - D(t,a)}{w} \tag{4.1}$$

- Exponential distance (expDist):

$$expDist(t,a) = \frac{1}{D(t,a)} \tag{4.2}$$

- Log distance (logDist):

$$logDist(t,a) = \log_2\left(1 + \frac{1}{D(t,a)}\right) \tag{4.3}$$

where:
$D(t,a)$: the word distance between the term t and the aspect a in the current passage of the document.

w: the predefined size of the word window before and after the aspect occurrence (currently 5). This normalizes the linear distance values between 0 and 1.

linDist can only be used in the Word window method because it requires the predefined word window size to normalize the weighting between 0 and 1. expDist and logDist can be used in both the Word window and Whole document methods because they only require the word distance to calculate the weighting.

4.1.3 Phrase support

The extension of the two methods above to identify phrases within the documents is now described. First, we need a limit of the phrase length. This is currently set to 3, which means phrases can only be a maximum length of 3. At each position where a term would is added in both the word window and whole document methods above, that term is also combined with its following term and added, It is ranked using the same method that is used for the single term. Phrases are built from a length of 2 up until the maximum phrase length for each position in the document where a single term would be added. Because a sorted set data structure is used, phrases that do not occur commonly enough will simply fall to the bottom of the set, and popular phrases will rise to the top. We will see the results of adding phrase support in the results of the following experiment.

4.2 Experiment: Comparing distance functions

The goal of this experiment is to see if distance functions can produce better vocabularies than the document frequency method used in AbraQ.

The following term-ranking methods will be tested:

- AbraQ: document frequency/co-occurrence strength
- Linear distance (linDist)
- Exponential distance (expDist)
- Log distance (logDist)
- Linear distance/co-occurrence strength (linDistCS)
- Exponential distance/co-occurrence strength (expDistCS)
- Log distance/co-occurrence strength (logDistCS)

4.2.1 Overview

Since there is no way of officially determining whether or not a term is relevant to an aspect, this is evaluated by manually labeling each term as relevant or irrelevant. Terms that are considered relevant are ones that relate to the aspect in a general sense. This means that any terms that are related to the aspect in some specific way are considered irrelevant. For the aspect "attacks" for example, the term "heart" (as in "heart attacks") would be considered irrelevant, whereas the term "kill" would be considered relevant.
The order in which the vocabulary terms are ranked is not considered, only the number of relevant terms are considered. Therefore, a better vocabulary will simply be measured

by the fraction of terms that are manually classified as relevant out of the total number of terms in the vocabulary. For example, the AbraQ method may produce 28/50 relevant terms where the expDist method may produce 32/50 relevant terms, in which case the expDist method will be the better method. The method of combining the distance function with the co-occurrence strength is the same as AbraQ: The distance function will be applied replacing document frequency to retrieve the top 200 terms, then these terms will be reordered using the co-occurrence strength and the top 50 terms will be retained. The Whole document retrieval method is used for all methods except the linDistMI method, in which case the Word window method will be used.
The experiment is performed on 7 aspects: "Black bear", "Attacks", "Abuse", "Email", "International", "Art" and "Crime".

4.2.2 Results and discussion

Table 4.2 shows the percentage of relevant terms out of the top 5, 10 and 50 for each method. A summary of the top 3 terms for each method for each of the 7 aspects is shown in Table 4.3.

	AbraQ	**linDist**	**expDist**	**logDist**	**linDistCS**	**expDistCS**	**logDistCS**
Top 5	37%	34%	34%	40%	34%	43%	43%
Top 10	39%	31%	34%	34%	37%	41%	40%
Top 50	19%	19%	20%	19%	21%	22%	24%

Table 4.2: Percentage of relevant terms out of the first 5, 10 and 50 terms for each vocabulary building method

In general, combining the distance-function with the co-occurrence strength has shown to improve the quality of the vocabulary terms, where the precision was increased by around 5% on average.

Overall, this experiment gives evidence that distance functions are preferable when initially ranking terms, over using document frequency. The disadvantage in using distance functions over the document frequency method is that it is more inefficient to use the distance functions. The document frequency method requires one pass over each document in the result set, whereas the distance functions require two passes. The first pass finds the occurrences of the aspect, and the second pass adds each term in the document to the vocabulary, weighted based on the closest occurrence found in the first pass. Because the goal of this project is to improve the quality of the expanded queries, efficiency is a minor issue because the goal is to obtain better quality results.
These results show that the distance functions without the co-occurrence strength perform almost as well as the AbraQ method. logDist is better than the AbraQ precision, because there are slightly more relevant terms ranked higher, but the matching Top 50 values indicate that each vocabulary has the same number of relevant terms. The fact that logDist performs about the same as AbraQ is in itself an improvement, because no co-occurrence strength was required to re-rank the terms. The use of the co-occurrence strength makes the algorithm less efficient, because two search queries are required for each of the 200 terms to retrieve the search counts, resulting in 401 total queries.

We can also see that the addition of phrase support is beneficial in the results in Table 4.3. Some useful phrases that have been identified are: "ursus americanus" for the aspect: "black bear", "organized crime" for the aspect: "crime", and "visual arts" for the aspect

BLACK BEAR						
AbraQ	**linDist**	**expDist**	**logDist**	**linDistCS**	**expDistCS**	**logDistCS**
ursus: 0.566	bear: 0.128	bear: 0.117	bear: 0.150	ursus americanus: 0.959	ursus americanus: 0.960	ursus americanus: 0.954
grizzly: 0.106	black: 0.070	black: 0.069	american: 0.075	ursus: 0.029	ursus 0.029	ursus 0.029
wildlife: 0.052	american: 0.055	american: 0.066	black: 0.073	wildlife: 0.003	wildlife: 0.003	grizzly: 0.005
ATTACKS						
AbraQ	**linDist**	**expDist**	**logDist**	**linDistCS**	**expDistCS**	**logDistCS**
police: 0.200	service: 0.068	service: 0.076	service: 0.081	anxiety panic: 0.989	anxiety panic: 0.925	anxiety panic: 0.925
north: 0.125	panic: 0.062	panic: 0.071	panic: 0.073	ddos: 0.006	panic attacks: 0.034	panic attacks: 0.034
killing: 0.071	september: 0.061	september: 0.067	september: 0.065	bombings: 0.004	bin laden: 0.019	bin laden: 0.019
ABUSE						
AbraQ	**linDist**	**expDist**	**logDist**	**linDistCS**	**expDistCS**	**logDistCS**
alcohol: 0.594	child: 0.091	child: 0.096	child: 0.106	alcohol: 0.270	alcohol: 0.270	alcohol: 0.272
sexual: 0.093	physical: 0.048	physical: 0.049	physical: 0.050	substance: 0.195	substance: 0.195	substance: 0.195
child: 0.049	sexual : 0.035	sexual: 0.037	sexual: 0.040	spousal: 0.147	spousal: 0.147	spousal: 0.147
EMAIL						
AbraQ	**linDist**	**expDist**	**logDist**	**linDistCS**	**expDistCS**	**logDistCS**
targeted: 0.138	etiquette: 0.035	mail: 0.030	address: 6.724	address: 5.557	address: 6.724	address: 3.386
services: 0.173	mail: 0.028	etiquiette: 0.027	history: 0.000	receive real: 0.000	history: 0.000	history: 0.000
inbox: 0.099	address: 0.026	web: 0.026	top: 0.000	history: 0.000	top: 0.000	top: 0.000
ART						
AbraQ	**linDist**	**expDist**	**logDist**	**linDistCS**	**expDistCS**	**logDistCS**
fine: 0.108	crimes: 0.022	crimes: 0.027	work: 0.023	prints: 0.121	visual arts: 0.164	prints 0.124
museum: 0.106	fine: 0.020	work: 0.021	fine: 0.022	framed: 0.113	prints: 0.100	framed: 0.115
exhibitions: 0.078	work: 0.019	fine: 0.021	artists: 0.020	fine: 0.108	framed: 0.092	fine: 0.110
CRIME						
AbraQ	**linDist**	**expDist**	**logDist**	**linDistCS**	**expDistCS**	**logDistMI**
police: 0.272	art: 0.037	art: 0.040	art: 0.044	victimless: 0.435	victimless: 0.344	victimless: 0.432
rape: 0.201	criminals: 0.023	book: 0.023	law: 0.023	true: 0.107	organized crime: 0.155	true: 0.107
victim: 0.115	book: 0.022	magazine: 0.022	book: 0.021	violent: 0.095	true: 0.085	violent: 0.075

Table 4.3: Top 3 terms produced by AbraQ, distance-functions and distance-functions combined with co-occurrence strength for different aspects. Each term is shown with it's weighting.

"art". The nature of using sub-queries for vocabulary terms is that many of the terms are related to some specific topic that includes the aspect. For example, when searching "attacks" to find terms, documents are often returned that relate to different topics, such as "heart attacks", "panic attacks" or "september 11 attacks". Such terms were ranked as irrelevant and this explains the low precisions in general, where all of the methods produced less than half relevant terms.

The method that will be used in AbraQNew is logDistCS.

4.3 Experiment: Increasing number of documents analysed in sub-queries

This section investigates the quality of the vocabularies generated by sub-queries, by varying the amount of documents analysed in each sub-query. AbraQ analyses 10 documents, but [4] claims that improving the quantity would improve the vocabulary quality.

4.3.1 Overview

This experiment compares vocabularies for 3 aspects from 3 individual queries. The number of documents that will be tested are: 5, 10, 20, 50 and 100. Table 4.4 shows the top 10 vocabulary terms for each aspect. The ranking method used is linDist with a word window limit of 5, and the retrieved terms are limited to single words for simplicity.

4.3.2 Results and discussion

	Top documents				
Aspect	**Top 5**	**Top 10**	**Top 20**	**Top 50**	**Top 100**
Abuse	1: abusing 2: net 3: child 4: sexuality 5: people	1: abusing 2: physical 3: child 4: net 5: verbal	1: child 2: net 3: abusing 4: physical 5: sexuality	1: child 2: abusing 3: abusing 4: physical 5: children	1: child 2: neglect 3: prevention 4: abusing 5: children
Attacks	1: service 2: denial 3: september 4: united 5: dos	1: service 2: denial 3: september 4: anxiety 5: panic	1: service 2: denial 3: september 4: news 5: ddos	1: panic 2: service 3: denial 4: networks 5: anxiety	1: panic 2: service 3: anxiety 4: networks 5: denial
Black bear	1: north 2: brown 3: american 4: states 5: wildlife	1: north 2: american 3: brown 4: states 5: areas	1: north 2: cub 3: foods 4: american 5: areas	1: north cub 3: homes 4: areas 5: inn	1: hunting 2: inn 3: homes 4: wildlife 5: north

Table 4.4: Top terms found for 3 aspects, using sub-queries with a range of document counts

It appears that the increase in documents analysed seems to have a minor improvement on the quality of the top terms. Terms in the top 100 have a more general relation, where terms from the top 5 documents seem to relate to more specific instances of the aspect. For example, in the top 5 documents for "abuse", sexuality relates to sexual abuse. However, in the

top 100 documents, sexuality is filtered out and "neglect" and "prevention" have appeared, which are arguably more general terms. Specific terms are less useful for an aspect's vocabulary simply because it is less likely to relate to all usages of the aspect. There are still specific terms that have remained even with a varied document count. For "abuse": "child" has remained in the vocabulary. For "attacks": "denial" has remained, where "denial" is related to the "Denial-Of-Service" computer attack, a specific topic of "attacks".

In summary, increasing the document count has made a minor improvement, where the terms are somewhat more general with more documents analysed. Increasing the document count in the algorithm from 10 to 100 would greatly slow the efficiency of the algorithm. In the worst case, 90 documents would need to be analyzed for each sub-query, if the document sets were independent. Although the focus of this project does not concern efficiency, it would greatly slow the process of testing the algorithm throughout the project if the document count was set to 100. The document count is only increased to 20 for AbraQNew.

4.4 WordNet

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (Synsets), each expressing a distinct concept [10]. Using a WordNet database has the advantage that it is much more efficient, and provides a list of synonyms that have strong semantic relationships. The disadvantage is that it is static, so does not provide as up to date information compared to LDA and GDA techniques, which leverage the dynamic content of web pages.

Some examples of synonyms returned by a WordNet database are shown in Table 4.5. To retrieve a Synset, the term and it's Part-of-Speech type are required. This is determined by using a Part-of-Speech tagger to tag each aspect in the query. In general, the synonyms from a WordNet are good quality terms compared to the terms using Sub-Queries, because they have been constructed using human input.

Aspect	**Type**	**Synonyms**
Attack	Noun	Onslaught, Onset, Onrush
Attack	Verb	Assail
Abuse	Noun	Maltreatment, Ill-treatment, Ill-usage
Black bear	Noun	Asiatic black bear, Ursus thibetanus, Selenarctos thibetanus
Crime	Noun	Offense, Criminal Offense, Law-breaking

Table 4.5: Synonym sets returned by a WordNet database

The WordNet terms will be added to the new vocabulary. It is hard to determine the usefulness of WordNet terms in the vocabulary. Because the WordNet synonym sets are constructed using human input, it can be assumed that the addition of WordNet terms will improve the quality of the vocabulary, and will not hurt the quality because the WordNet synonyms have a predetermined semantic relationship. Chapter 6 will detail the experiments to evaluate the query expansion performance using the new vocabulary enhanced by WordNet.

4.5 Conclusions

In this chapter, the vocabulary building component of AbraQ has been improved. The set of sub-queries that are generated have been slightly modified to compensate for the likelihood

of incorrect aspect identification. It has been discovered that the increase in the number of documents analysed from the result set of sub-queries improve the quality of the vocabulary, by yielding more generalised vocabulary terms, and it was decided that the number will be increased from 10 to 20 for AbraQNew. Document frequency is used in AbraQ to retrieve an initial list of 200 terms from sub-queries, and this list is refined to 50 using a co-occurrence strength. However, the experiment in 4.2 showed that a better quality set of terms can be retrieved when using the logged distance function to retrieve the initial 200 terms from the sub-queries. The addition of phrase support has also improved the quality of the vocabulary. WordNet terms have been added to the new vocabulary, where they are ranked higher than the terms originating from the sub-queries. It is assumed that they are consistently higher quality terms than the sub-query terms.

Chapter 5

Query-Document relevance classification

It is useful to find a way of determining whether or not a document is relevant to a search query. This is useful for the following reasons:

- Determining whether a query requires expansion, based on whether a high enough frequency of documents are relevant.
- Finding good quality terms for expansion using the relevant documents only, reducing the likelihood of query drift.
- Determining whether an expanded query performs better than the original one.

For a given hard query, 2 out of 10 documents may be relevant. We know that local document analysis provides a useful source for expansion terms when the documents are relevant. Therefore, if we can identify the 2 relevant documents, then we are able to use terms from these documents and reduce the chance of hurting the query with more irrelevant terms. This chapter discusses a new method of determining the relevance of a document in accordance with a search query. The method is compared against a baseline of manually determined relevance to gauge the performance.

As mentioned in 2.3.4, AbraQ measures the relevance of an expanded query by summing the aspect scores across the set of documents. In addition to how well the collection of aspects are covered in the document set, it is also important to determine how balanced the coverage of the set of aspects are. AbraQ also does not identify the relevance of specific documents from the query, so the ability to determine relevance of individual documents is an improvement.

[4] mentions that a good refinement is a query that produces a result set with no under-represented aspects. This idea can be used to identify how relevant a particular document is to a query, by basing the measurement of relevance on how evenly the aspects are represented in the query.

5.1 Algorithm

We need to measure the relevance of a document from a search result set against a query. The inputs to the algorithm are the query and the contents of a particular document. The output is true or false, depending on whether that document is relevant/not relevant. In the case that we are finding the relevance of a document from the original query, we simply

give the original query and the document contents. When we want to find the relevance of a document from an expanded query, we give the original unexpanded query and the document that resulted from the expanded query. This is because we are comparing the document to the original query. That is, the set of aspects and associated vocabularies from the original query. This allows us to determine whether or not the set of documents from the expanded query satisfy the vocabularies of the original query better than the documents from the original query.

5.1.1 Aspect scores

First, we need to calculate a set of aspect scores, where each score is a number that determines how well the document covers an aspect in the query.

For a document d, an aspect a and its vocabulary v:

$$aspectScore(d, a) = \frac{\sum_{i=1}^{N} occurrences(v_i, d) * weight(v_i)}{length(d)} \tag{5.1}$$

N: the number of terms in v

$occurrences(v_i, d)$: The number of occurrences of the stemmed version of a term v_i in the stemmed version of the document d

$weight(v_i)$: The weighting of v_i in the vocabulary v

$length(d)$: The number of words in a document d

Each aspect score for a document should always be approximately between 0 and 1, because it is normalized by the word length of the document. It would only be equal to 1 in the highly unlikely case that every term in the document was a term in the aspect's vocabulary, and each term's associated weight is 1.0. It is assumed that this will not occur because the documents returned by Google will contain content that makes sense, instead of a pure list of keywords. Documents that contain valid sentences will contain stop words such as: "and, this, the, or", which can never make an aspect's vocabulary, so it is unlikely that every term in the document will be counted as a vocabulary term occurrence.

We can then use the set of aspect scores to calculate whether or not d is relevant.

5.1.2 Document labelling

A relevant document contains an even coverage of aspects, while each aspect is covered well. If one aspect is covered very well and another is covered barely, then the document is not relevant because it does not contain a balance of information for each aspect that the user requested in their search query. Also, it is not enough to check that the aspect coverage is balanced if all of the aspects are not covered well at all, in which case the aspect scores would be all close to zero.

To measure how balanced the set of aspect scores are, we can use the standard deviation of the aspect scores, where a lower standard deviation is better because the aspect scores vary less. We need to define a threshold that the standard deviation of the aspect scores has to be less than to be valid.

To measure how well the aspects are covered, we can use the average of the aspect scores, where a higher average is better because the aspects are covered better. We need to define a threshold that the average of the aspect scores has to be greater than to be valid.

The following formula labels a document as relevant or irrelevant based on the set of aspect scores:

$$documentIsRelevant(d,a) = \left(\sigma_{aspectScores(a)} \leq SD\right) \wedge \left(\mu_{aspectScores(a)} \geq A\right) \quad (5.2)$$

$aspectScores(a)$: the set of aspect scores for all aspects in d
SD: A predefined threshold for the standard-deviation
A: A predefined threshold for the average

The next section experiments with the standard deviation and average thresholds to find the best thresholds to use.

5.2 Experiment: Finding optimal standard deviation and average thresholds

This experiment compares nine different combinations of standard deviation and average thresholds to classify the relevance of a document. The top 30 documents from five queries in the TREC 2005 HARD Track [17] are used. The relevance of each document has been determined manually and this is used to gauge how well each set of thresholds perform. The relevance is determined in accordance with the TREC descriptions that accompany the queries.

5.2.1 Results and discussion

Overview

Table 5.1 shows the classification accuracy for the range of weights for each of the 5 queries that were tested. Table 5.2 shows an average of the number of false positives, documents that are determined relevant using the automatic method which are not relevant in the manual method, and false negatives, documents determined not relevant using the automatic method which are relevant in the manual method.

Interpretation

A is the value of the average document aspect score threshold, and SD is the value of the average document standard deviation score threshold.
For example, in Table 5.1, where A = 0.6, SD = 0.5: A relevant document is classed as relevant when the average of the aspect scores for that document is ≥ 0.6, and the standard deviation of the set of aspect scores for that document is ≤ 0.5. The value of 70.0% for "Black bear attacks" indicates that the combination of these two particular thresholds resulted in the relevance algorithm correctly classifying 70.0% of the 30 documents. The accuracy of classification is compared to the manual classification.

Query	SD: 0.5			SD: 0.3			SD: 0.1		
	A: 0.6	A: 0.7	A: 0.8	A: 0.6	A: 0.7	A: 0.8	A: 0.6	A: 0.7	A: 0.8
Black bear attacks	70.0%	76.7%	76.7%	63.3%	70.0%	70.0%	63.3%	70.0%	70.0%
Abuses of email	70.0%	73.3%	70.0%	73.3%	76.7%	73.3%	73.3%	76.7%	73.3%
Airport security	80.0%	80.0%	83.3%	80.0%	80.0%	83.3%	80.0%	80.0%	83.3%
Wildlife extinction	76.7%	76.7%	76.7%	76.7%	76.7%	76.7%	76.7%	76.7%	76.7%
Cult lifestyles	100.0%	96.7%	96.7%	100.0%	96.7%	96.7%	100.0%	96.7%	96.7%
Average	**79.3%**	**80.7%**	**80.7%**	**78.7%**	**80.0%**	**80.0%**	**78.7%**	**80.0%**	**80.0%**

Table 5.1: Classification accuracy for the top 30 documents in 5 queries, using different average and standard deviation threshold values.

Query	SD: 0.5			SD: 0.3			SD: 0.1		
	A: 0.6	A: 0.7	A: 0.8	A: 0.6	A: 0.7	A: 0.8	A: 0.6	A: 0.7	A: 0.8
False positives	9.61%	12.0%	8.7%	11.3%	9.3%	8.0%	11.3%	9.3%	8.0%
False negatives	10.43%	8.7%	10.7%	10.0%	10.7%	12.0%	10.0%	10.7%	12.0%

Table 5.2: Average classification error rates.

Discussion

The algorithm performed reasonably well over set of documents in the 5 queries that were tested. In the range of weights tested, the average false positive rate was 9.72% and the average false negative rate was 10.58%. It is more important to focus on weightings with a lower false positive rate, as it is better to filter out more documents in order to increase the chances of retrieving more relevant ones. Therefore, the best weighting found was the standard deviation of 0.3 and average of 0.8. This produced a false positive rate of 8% and a false negative rate of 12%.

5.3 Conclusions

The document relevance classification algorithm is fundamental to the success of the Automatic Query Expansion algorithm. There is no perfect combination of weights to perform this task, and the algorithm is not able to determine the relevance 100% of the time. This is because the relevance of a document really depends on whether or not the user who wrote the search query finds it useful.

The algorithm described in this chapter has shown to classify document relevance reasonably accurately, when compared to manual human classification. The weakness is that it depends heavily upon the previous stages of the algorithm: determining the correct aspects and building a vocabulary of terms that are strongly related to those aspects. However, a reliable mechanism to identify aspects and for building vocabularies is hard to implement, because there is much uncertainty as to what should be classified as a valid aspect, and what terms are defined as strongly related to those aspects.

Because there is a lot of grey area in what is relevant and what is not, the best we can do programmatically is to filter out documents that appear to have content irrelevant to the user's search query. This algorithm shows a reliable method of performing this task because it uses vocabularies to determine whether or not the document contains likely keywords that the user is looking for, other than the keywords used in the query itself. The classification accuracy of the algorithm is highly dependent on the accuracy of the aspect identification and the quality of the vocabularies.

Chapter 6

Query expansion

It is now possible to determine with reasonable confidence, the relevant documents to the original query and the documents from the expanded query that are relevant to the original query. The set of aspects in the query, their vocabularies, and how well each aspect is covered in each document of the query are already known, so the final component of the algorithm is to find the best method of producing the expanded query. An experiment will investigate a range of sources for expansion terms. Once the best set of sources are selected, the performance of AbraQNew will be tested against AbraQ and other query expansion algorithms.

First, some brief implementation details are described.

6.1 Implementation details

A query is only expanded if the number of relevant documents from the original query are less than a predefined threshold, which is currently set to 70.0%. This decision is made because the aim is to improve hard queries, where typically less than 70% of the documents are relevant. Once we have determined that a query requires expansion, the aspect with the lowest average aspect score in the top 10 documents is selected as the underrepresented aspect. It is better to expand queries with a lower number of terms, because adding too many terms can hurt the precision of the query. Therefore, only one underrepresented aspect is selected for expansion.

6.2 Experiment: Expansion term sources

This section investigates the best terms to use when expanding the query. A number of sources exist for expansion terms for an under-represented aspect. These are:

1. The top vocabulary terms for the aspect
2. The WordNet synonyms for the aspect
3. The terms co-occurring with the aspect in the relevant documents of the top 10 documents from the original query.
4. The terms co-occurring with the aspect in the irrelevant documents of the top 10 documents from the original query

The goal is to find the sources that produce the best terms, so we can use only the best sources to produce our expanded set of queries. If Source 3 performs consistently better than Source 4, it is evident that the added ability to determine relevant documents in the original query is an improvement over AbraQ.

To retrieve the terms from the top documents from the original query. The logDist method (described in 4.3) is used with the Whole document method. It is assumed that the terms from these documents are already highly relevant so no co-occurrence strength is applied to rerank the term order.

6.2.1 Overview

For simplicity, this experiment is performed on queries with only one underrepresented aspect and each expanded query only uses one expansion term.

The experiment is run over 10 queries. For each query, a set of expanded queries are produced for each source described above. Sources 1, 2 and 4 are limited to the top 10 weighted terms, so produce 10 expanded queries each. The WordNet synonyms (Source 3) produce an expanded query for each synonym from the first sense from the Synset of the underrepresented aspect. Each expanded query is produced by using a term from each sources term set, and appending it after the under-represented aspect in the query. For the set of expanded queries for each source, the quality of that source is measured using the average percentage of relevant documents produced by all the expanded queries out of the total number of documents produced.

Since the vocabulary contains terms from the WordNet, only the terms that result from sub-queries will be tested for Source 1, and the WordNet terms will be tested in Source 2.

6.2.2 Results and discussion

Query	**WordNet**	**Vocabulary**	**Irrelevant docs**	**Relevant docs**
Black bear attacks	22.50%	9.00%	41.00%	51.11%
Abuse of email	5.00%	0.00%	0.00%	9.00%
International art crime	12.50%	4.00%	23.00%	31.11%
Nobel prize winners	55.00%	18.00%	18.00%	16.25%
Airport security	-	3.00%	1.00%	-
Mental illness drugs	50.00%	15.00%	32.00%	42.00%
Automobile recalls	22.50%	5.00%	31.00%	32.00%
Overseas tobacco sales	20.00%	20.00%	11.00%	15.00%
Wildlife extinction	-	12.00%	22.00%	29.00%
Inventions, scientific discoveries	-	1.00%	5.00%	3.75%
Average	26.79%	9.33%	18.40%	25.47%

Table 6.1: Average percentage of relevant documents for resulting from expanded queries, produced using a range of expansion sources.

Table 6.1 shows that the WordNet produces the best result, although the performance is inconsistent across the query set, where the average percentage of varies between 5.00% and 50.00%. The top relevant documents produce the next best results, followed by the top irrelevant documents. It is clear from these results that the use of the top relevant documents performs consistently better than using the set of top irrelevant documents. This is one improvement over AbraQ, which uses all the top documents without distinguishing between

relevant and irrelevant documents. One thing to note about these results is that the expansion is harder when the original query produces less relevant documents, so queries such as "Airport security" have produced poor results because there was no relevant documents, compared to "Black bear attacks" which had 40.00% relevant documents. The vocabulary for the underrepresented aspect performs very poorly, producing a maximum of 20.00% of relevant documents.

Each source could not always be utilized for each query. This occurred when using the WordNet on the queries: "Wildlife extinction" and "Inventions, scientific discoveries". This is because there were no terms in the WordNet synonym set for both "extinction" and "scientific". It also occurred when queries did not produce any relevant documents, such as the query "Airport security".

Query	**Best expansion**	**Expansion source**
Black bear attacks	Black bear attacks north	Top relevant docs
Abuse of email	Abuse of email spam reporting	Top relevant docs
International art crime	International cultural art crime	Top relevant docs
Nobel prize winners	Nobel prize winners nominations	Vocabulary
Airport security	Airport security summary	Top docs
Mental illness drugs	Mental illness drugs abuse	Top relevant docs
Automobile recalls	Automobile vehicle safety recalls	Top relevant docs
Overseas tobacco sales	Overseas tobacco sales body	Top relevant docs
Wildlife extinction	Wildlife threatened extinction	Top relevant docs
Inventions, scientific discoveries	Inventions world, scientific discoveries	Top irrelevant docs

Table 6.2: Best expanded queries and their expansion source.

Table 6.2 shows the best expanded query for each query, out of all the expansion sources tested. It is clear that the best source from these results is the top relevant documents. However, there are still instances where top irrelevant documents produce the best expansion such as in the queries: "Airport security" and "Inventions, scientific discoveries". The likely cause of this is because the relevance detection algorithm is strict, in that there are less false positives than false negatives, so there is still a chance that a few relevant documents may be classified as irrelevant.

6.2.3 Summary

The results of this experiment indicate a useful order for expansion sources.

This order will be:

1. Documents labelled as relevant
2. Documents labelled as irrelevant
3. WordNet
4. Vocabulary

That is, expanded queries will be produced, where the first queries will use terms from the top documents of the original query, before using terms from WordNet, and so on.

6.3 Experiment: Automatic comparison of AbraQ, AbraQNew and traditional systems.

This experiment will compare the resulting system AbraQNew with AbraQ and other traditional query expansion algorithms. The goal is to find out if AbraQNew outputs an expanded query of better quality than the other systems.

6.3.1 Traditional expansion systems

Below is a summary of the traditional query expansion systems that are compared in the experiment. Each system ranks terms using the top 10 documents from the result set of the original query. If there are more than one term that share the highest ranking, one will be chosen from random for each query that is expanded.

Document frequency (DF)

Terms are ranked by the number of documents that they occur in. If a particular term x occurs in 7 out of the 10 documents, and another term y occurs in 5 out of the 10 documents, then x will be the highest ranked term.

Term frequency (TF)

The term frequency system ranks terms by the number of times they occur out of the total number of terms. The implementation in this experiment simply treats the top 10 documents as one large document. The top terms are the ones that occur the most frequently in these top 10 documents.

Term frequency-Inverse document frequency (TF-IDF)

This system ranks by term frequency as described above, but then multiples each term ranking by an inverse document frequency value. The inverse document frequency is described as a measure of the general importance of the term. It is calculated by dividing the total number of documents in a corpus by the number of documents that contain the term.

6.3.2 Overview

This experiment will compare the amount of additional relevant documents produced by the best expanded query from each algorithm against the amount of relevant documents produced by the original query. 50 queries are tested, where each algorithm produces an expanded query. The first 45 queries are from the TREC 2005 HARD Track [17]. All queries greater than one word were selected because the algorithm cannot currently process single word queries. The last 5 queries are selected at random from the TREC 2003 HARD Track [18]. The relevance detection algorithm from AbraQNew is used to find the number of relevant documents by the expanded query generated by the other systems.

6.3.3 Results interpretation

Tables 6.3 & 6.4 show the 50 queries that were expanded by each system. Each row contains a query along with the percentage of relevant documents classified using the document classification algorithm for the original query. Following this, the expansion term produced

by each system and the percentage of relevant documents that resulted the expanded query that used that expansion term is shown.

Any queries that AbraQ or AbraQNew left alone are shown by the "-" symbol. Such queries were left alone when no better expansion could be found. This could be because:

1. All the aspects are represented in the case of AbraQ
2. There were at least 70% relevant documents returned in the case of AbraQNew
3. None of the top ten expanded queries were able to produce an expanded query that resulted in a higher number of relevant documents than the original query

6.3.4 Discussion

These results show that AbraQNew performs consistently better over AbraQ. In most cases that AbraQ generated an improved query, AbraQNew produced a better or matching query (No. 1, 2, 5, 12, 14, 16, 19, 26, 27, 32, 37, 39, 41, 43, 44, 45, 46, 47, 49). Occasionally, AbraQNew was unable to produce a better query than AbraQ (No. 10). The addition of phrase identification has produced useful expansion terms, such as "antarctic explorers" (No. 14), "car tires" (No. 26) and "work dog" (No. 27).

For query No 45: "U.S., investment, Africa" had a precision of 20%. AbraQNew claimed that the addition of "United States of America" to support "US." resulted in 100% precision. On manual inspection, it was noted that the original query had 30% precision and the expanded query: "U.S. United States of America, investment, Africa" had 80% precision. The addition of this phrase produced documents that had information on US investment but contained "United States" more frequently. This expansion was one of the best expansions produced by AbraQNew and the expansion phrase originated from WordNet, which gives evidence that WordNet can occasionally produce very high quality expansion terms.

Query No 49: This query exhibits query-drift occurring in AbraQNew. The term "california" is produced for the query "recent earthquakes". Such a term would narrow the search down to information about earthquakes in only that area. It may be coincidental that such a term produces more relevant documents, as there happens to be more documents related to "recent earthquakes" that concern "california".

Experiment issues

It is important to note that these experiment results are produced by the document relevance classification algorithm, which creates bias toward AbraQNew. This is because AbraQNew also uses the same document relevance classification algorithm to find the best query for each expansion algorithm. AbraQNew generates 10 expanded queries and chooses the one that performs the best. If it can not find an expanded query that improves the number of relevant documents, no query is generated. The other algorithms return the best expanded query, which is checked using AbraQNew's document relevance classification algorithm.

The document relevance classification algorithm did not always perform accurately in the above results, in which case it would identify a lower number of relevant documents than were in the set. This is caused by bad vocabularies, a bad set of terms that do not identify content in relevant documents. A good example of this is seen in the query: "Hubble telescope achievements". It classified all of the top 10 documents as irrelevant, when on manual inspection there were 4 relevant documents. Poor vocabularies were generated for both "hubble telescope" and "achievements". Some of the top terms for "achievements" are listed in Table 6.6.

No.	Query	DF	TF	TF-IDF	AbraQ	AbraQNew
1	black bear attacks 30%	wildlife 10%	bear 0%	campsite 40%	killed 50%	north 50%
2	international art crime 40%	art 40%	art 40%	charney 60%	support 30%	experts 50%
3	abuse of email 30%	report 30%	message 10%	uia 0%	programs 10%	-
4	hubble telescope achievements 0%	space 0%	player 0%	hierarchies 0%	play 0%	-
5	new hydroelectric projects 0%	years 20%	search 10%	bickle 0%	small 0%	consciousness 20%
6	airport security 30%	privacy 0%	page 10%	favorite 0%	flights 0%	-
7	nobel prize winners 0%	medicine 0%	images 0%	improb 0%	-	-
8	mental illness drugs 70%	informed 10%	health 40%	pharmacotherapy 30%	-	-
9	home schooling 0%	homeschool 0%	learning 0%	legalities 0%	-	-
10	automobile recalls 60%	type 50%	car 90%	alldatadiy 0%	official 80%	-
11	three gorges project 0%	dam 0%	river 0%	cofferdam 0%	china 0%	-
12	wildlife extinction 60%	species 60%	google 10%	pagead 30%	fish 60%	plants 70%
13	inventions, scientific discoveries 20%	type 0%	text 20%	pagead 0%	-	greatest 20%
14	antarctica exploration 50%	world 70%	south 50%	shackleton 10%	real 50%	antarctic explorers 70%
15	journalist risks 0%	home 0%	commentators 0%	flogging 0%	story 0%	-
16	human smuggling 30%	transportation 50%	google 0%	sevp 0%	financial 30%	trafficking 50%
17	transportation tunnel disasters 30%	information 30%	states 10%	immersed 20%	include 10%	-
18	hydrogen energy 70%	planned 40%	carbon 40%	kwinana 10%	-	efficient 90%
19	illegal technology transfer 50%	related 30%	page 60%	caplan 20%	day 30%	intellectual 60%
20	mercy killing 0%	medical 0%	medical 0%	kevorkian 0%	-	-
21	native american casino 60%	years 60%	indian 80%	tulalip 20%	-	indian 80%
22	oceanographic vessels 50%	research 50%	oceanography 40%	corwith 20%	coast 20%	-
23	foreign minorities, Germany 30%	make 10%	cultural 30%	interscience 0%	-	policy 30%
24	Ireland, peace talks 70%	irish 50%	page 50%	loyalist 50%	-	party talks 70%
25	tropical storms 60%	wind 40%	forecast 50%	nmi 50%	-	-

Table 6.3: Query precisions of each query expansion system. First 25 queries.

No.	Query	DF	TF	TF-IDF	AbraQ	AbraQNew
26	recycle, automobile tires 0%	products 0%	text 0%	pagetracker 0%	high 0%	car tires 10%
27	law enforcement, dogs 60%	police 70%	work 70%	vscript 20%	left 10%	work dog 80%
28	UV damage, eyes 100%	protection 80%	rays 100%	macula 90%	color 70%	-
29	curbing population growth 40%	world 30%	rightful 20%	rightful 40%	natural 40%	-
30	microsoft monopolies 50%	share 40%	windows 40%	pagead 50%	-	developing 70%
31	ship losses 10%	time 0%	navy 10%	mutinied 10%	-	sunk 10%
32	price fixing 30%	sell 40%	competition 20%	cartels 10%	supply 10%	laws 40%
33	arrests bombing WTC 70%	terrorism 40%	attack 50%	createdat 0%	-	trade center 80%
34	wrongful convictions 0%	recent 0%	case 0%	yfbq 0%	case 0%	-
35	consumer online shopping 80%	making 80%	information 90%	ordered 90%	devices 70%	-
36	family leave law 90%	home 100%	employment 100%	fmla 100%	money 80%	-
37	tax evasion indicted 10%	home 0%	topix 0%	lrm 10%	pay 10%	jury 50%
38	U.S. ethnic population 80%	states 60%	americans 30%	cuban 20%	-	-
39	teenage pregnancy 50%	sexual 0%	pregnant 30%	leftnav 20%	mothers 30%	baby 50%
40	family-planning aid 0%	program 0%	reply 0%	leftnav 0%	visit 0%	-
41	iran-iraq cooperation 10%	countries 10%	countries 10%	larijani 10%	open 0%	security 30%
42	euro opposition 10%	policy 20%	search 0%	padding 0%	members 0%	-
43	legal, Pan Am, 103 10%	case 0%	article 20%	abdelbaset 10%	real 0%	scotland 10%
44	Greek, philosophy, stoicism 50%	stoics 60%	zeno 70%	indexicals 40%	centuries 40%	articles 80%
45	US., investment, Africa 20%	investors 70%	cca 70%	allafrica 10%	building 10%	United States of America 100%
46	animal protection 10%	news 10%	cat 20%	aspca 10%	safe 0%	organizations 20%
47	amusement park safety 50%	ride 50%	theme 70%	thrills 40%	injuries 30%	theme park 80%
48	y2k crisis 0%	working 0%	computer 0%	morella 0%	state 0%	-
49	recent earthquakes 60%	resources 90%	alaska 40%	koorda 20%	tsunami 20%	california 90%
50	mad cow disease 0%	prion 0%	click 0%	scrapie 0%	-	-

Table 6.4: Query precisions for each query expansion system. Second 25 queries

Original query	DF	TF	TF-IDF	AbraQ	AbraQNew
33%	30%	30%	19%	22%	54%

Table 6.5: Average precisions of the 50 queries for each query expansion system

Ranking	Vocabulary term
1	world warcraft
2	multiplayer achievements
3	unlocked
4	list
5	multiplayer

Table 6.6: Top vocabulary terms for "Achievements"

The problem here is the specific topics related to achievements that were found in the result set of the "achievements" query. This problem was not observed in the relevance detection testing because out of the 5 queries that were tested in 5.2, the vocabularies that were produced for the aspects for each query were of decent quality.

6.4 Experiment: Manual comparison of AbraQ and AbraQNew

AbraQ and AbraQNew are now compared using manual relevance classification, which is tested for all queries where both systems produced an expanded query. For each query, the top 10 documents from 1: the original query, 2: the expanded query produced by AbraQ and 3: the expanded query produced by AbraQNew are classified. These documents are classified using the descriptions accompanying the queries from the TREC 2005 HARD Track query list. The 3 queries from the 2003 TREC query list: "amusement park safety", "recent earthquakes" and "animal protection" are omitted from the results because the description for the first query was particularly strict and the descriptions for the latter two were out-of-date, so no relevant documents could be found for each query or the expanded queries.

Because of the biased results produced by the relevance detection algorithm, this experiment will provide more solid evidence that AbraQNew performs better quality expanded queries than the original query.

6.4.1 Results

The results in Table 6.7 show the precision of the document set produced by each original query. This is compared with the precision of the document set from the expanded queries that AbraQ and AbraQNew produced. The average precision is shown for the original queries, and for AbraQ and AbraQNew. The average improvement is shown for AbraQ and AbraQNew, which is the average additional number of relevant documents improved for each query that was improved by the system. The number of queries improved and queries worsened are shown for each system. A query is improved if it contains a greater number of relevant documents than the original, and a query is worsened if it contains a lesser number.

No.	Query	Original	AbraQ	AbraQNew
1	black bear attacks	30%	40%	50%
2	international art crime	0%	10%	10%
4	new hydroelectric projects	50%	0%	10%
12	wildlife extinction	20%	20%	40%
14	antarctica exploration	30%	30%	50%
16	human smuggling	20%	20%	20%
19	illegal technology transfer	20%	20%	10%
26	recycle, automobile tires	30%	20%	30%
27	law enforcement, dogs	30%	20%	30%
32	price fixing	10%	30%	0%
37	tax evasion indicted	70%	50%	60%
39	teenage pregnancy	60%	80%	60%
41	iran-iraq cooperation	70%	20%	70%
43	legal, Pan Am, 103	40%	50%	50%
44	greek, philosophy, stoicism	100%	70%	70%
45	us., investment, africa	30%	50%	80%
Average precision		38%	33%	40%
Average improvement		-	15%	21%
Queries improved		-	6	6
Queries worsened		-	6	5

Table 6.7: Manual relevance judgements of the top expanded queries produced by AbraQNew.

6.4.2 Discussion

Overall, the comparison of the best queries using manual classification has shown that AbraQNew performs better than AbraQ. Although the number of queries improved were equal, AbraQ worsened one more query than AbraQNew. A high number of worsened queries is a problem because the user would receive a greater number of irrelevant documents on average through the expanded query. Therefore the results show that AbraQNew is a slightly safer algorithm to use. Out of the 16 queries that were tested, 14 queries produced by AbraQNew were better than or equal to AbraQ, where 8 were better. AbraQNew showed a larger average improvement than AbraQ, meaning that the expanded queries that it produced were more effective, and resulted in a larger number of additional relevant documents on average. Query No 45 is one example of this. Both systems have shown to occasionally worsen the results of the original query. For example, for Query No 44, both systems have produced 7 relevant documents, where the original query produced 10.

There are some limitations of this experiment. Firstly, a small number of queries were tested and so the testing was not completely thorough due to the time constraints of the project. Secondly, we are limited to the descriptions specified by the TREC collection. All of the queries tested in these experiments consisted of 3 words or less. Since such a small amount of information is given to represent the search goal, the meaning of the search goal is very difficult to determine. Therefore, these descriptions are the best guide we have to follow. They help to prevent bias from personal opinions and allow a reasonably logical method of classification. Some issues with these descriptions are discussed in Section 6.4.3.

6.4.3 TREC query description issues

Specific descriptions

In general, the descriptions often specify that documents are relevant where, they contain instances or examples of the query topic, rather than documents that contain general explanations of the query topic. Typical queries with such descriptions are: "black bear attacks", "recent earthquakes", "illegal technology transfer", "price fixing" and "human smuggling". For example, the description and narrative for "human smuggling" from the TREC 2005 HARD Track query list is as follows:

```
Description:
    Identify incidents of human smuggling.
Narrative:
    A relevant document shows an incident of humans (at least ten)
    being smuggled.
    The smugglers would have to realize a monetary gain for
    their actions, while the people being smuggled may or may
    not be willing participants.
```

Listing 6.1: The TREC 2005 description and narrative for the query "Human smuggling"

The documents that appeared in Google for this query contained mostly information about human smuggling in general. A statistic such as "the amount of human smuggling that occurs annually in the USA" is a typical example, where such a document is classified as irrelevant. Although this issue is more related to the TREC descriptions, it is important to highlight because this is one of the main causes of the inconsistencies between the automatic and manual method.

Outdated descriptions

Although these results are slightly more reliable than the automatic relevance comparisons, the relevance is still not completely reliable. It is limited to the TREC 2003 and 2005 descriptions, which may have become out-of-date in 2009.

6.5 Conclusions

6.5.1 Expansion term sources

This chapter has investigated a range of sources for expansion terms. The experiment in Section 6.2 showed that the terms from the documents classified as relevant from the original query was the best source for expansion terms, followed by the documents classed as irrelevant. It seems unusual that the documents classified as irrelevant can perform better than the vocabulary terms. The reason that this is the case is because the irrelevant documents in the original query are likely to still contain terms more closely related to the query than terms from a sub-query, which is constructed from a subset of the aspects. The ability to use documents from the original queries document set is an improvement upon AbraQ. The only situation that AbraQ is able to utilize these documents is when the original query has two aspects. This is because AbraQ generates sub-queries from aspects and aspect pairs, and the original query is an aspect pair in this case.

The WordNet synonyms also proved to be a good expansion source occasionally, but its performance was inconsistent. The vocabulary built from sub-queries performed poorly in general. These results lead to the decision to utilize all the expansion sources tested, but in a defined order, with the best sources being utilized first.

6.5.2 Query expansion system comparison

The experiment in Section 6.3 compared the final AbraQNew system with AbraQ and other query expansion systems. AbraQNew showed the best performance. However, all systems struggled with some queries where there were no relevant documents. This experiment also highlighted that AbraQNew exhibits some query-drift, meaning that AbraQNew is not yet completely reliable. The following experiment in Section 6.4 compared the expanded queries produced by AbraQ and AbraQNew using manual relevance classification. This gave more reliable evidence that AbraQNew generally performs better than AbraQ.

Chapter 7

Conclusion

This project has made various improvements on the AbraQ algorithm to improve the quality of Automatic Query Expansion. The contributions of this project are first summarized. Following this, some limitations of the algorithm are described. Finally, areas of future work are proposed so the algorithm can be developed further.

7.1 Contributions

7.1.1 Aspect vocabulary improvement

The quality of the vocabulary terms retrieved from sub-queries was improved in AbraQNew. This was achieved through the use of distance functions that weight terms within top ranking documents based on their distances away from each aspect occurrence. This approach for initially retrieving terms helped to produce a better quality set than the document frequency ranking method used in AbraQ. Phrase support was added, which allowed for additional terms to be discovered that were longer than one word. The addition of WordNet as vocabulary terms also improves the vocabulary term quality, by using terms that have strong semantic relationships, which are predetermined by humans.

7.1.2 Relevance detection

This project has added the ability to detect specific documents that are relevant within a search query's result set. This has given us the ability to measure the quality of a result set, so we can automatically measure whether or not an expanded query's result set is of better quality than the original query's result set. This relevance detection ability also enabled us to identify which documents from the original query's result set to analyse for the purposes of selecting expansion terms.

7.1.3 Expansion improvement

AbraQNew, when directly compared to AbraQ and other traditional query expansion systems, has shown to make a considerable improvement in the quality of query expansion. The top ranking documents from the original query produce better quality expansion terms, than the terms retrieved from sub-queries in AbraQ. The addition of WordNet has also proven to make good quality expansion terms. Chapter 6 proved that the improvements made throughout the project has resulted in a better and more reliable query expansion algorithm than AbraQ.

7.2 Future work

This section describes some possible ideas for future development of AbraQNew.

7.2.1 Improve aspect identification

This section contains some possible ideas that could be attempted to improve the reliability of the aspect identification algorithm.

Optimize aspect threshold

As mentioned in 2.3.2, an aspect is valid if it's score (existence * support) is ≥ 1.0. Although this threshold of 1.0 works well, further investigation into finding an optimal threshold could be beneficial. This could be achieved using a supervised learning technique, through creating test sets and training sets of labelled aspect data. The value of 1.0 is arbitrary and attempting to optimize the threshold would increase the overall rate of correct aspect identification.

Local document analysis to identify aspects

The algorithm uses search counts to identify aspects. Because of this, aspects are often classified incorrectly when they occur less on the Internet. One such example is when searching from flights with airline companies. The algorithm identifies "Air New Zealand" as two aspects, "Air" and "New Zealand". However, it identifies "American Airlines" correctly as a single aspect.

It might be a more reliable method to analyse the documents from the original query when identifying the aspects. This would allow the aspect identification algorithm to identify the aspects within the context of the query. "Air New Zealand" is not identified correctly because there are less documents, but analysing the number of times that "Air New Zealand" occurs within the documents of the original query may help to resolve this issue.

7.2.2 Add support for multiple search engines

AbraQNew only uses the Google search engine. Using a different search engine could affect all stages of the algorithm. The aspect identification component would be affected by different search counts. The quality of the vocabularies would be affected, through analysing different document sets from sub-queries. The resulting expanded query could perform better or worse on other search engines. Google is known to produce very approximate search counts, so it may be beneficial to explore other search engines for this property.

7.2.3 Extend document relevance to site relevance

During the task of manually classifying the relevance of pages, it was noted that some documents were scored poorly, when they linked to other documents on the same website that were relevant pages. If the user clicks on such a site, the homepage may not be relevant because it does not have the content, but in fact the other pages linked to from the homepage can in fact contain relevant information. One example of this was observed during the search: "Nobel prize winners", one document in the results (at the URL: http://nobelprizes.com), was classified as not relevant, although the other pages on the site were very relevant to the user's search goal, where they contained lists of winners in different nobel prize categories. One area of improvement is to extend the document relevance

mechanism to take into consideration the pages linked to from a document for relevance, rather than just analysing the content of the document. This should be restricted to pages that share the same base URL as the original document. A possible method of considering the linked documents could be to half the original document score and allow the combined linked document scores to contribute to the other half, where each individual linked document score is normalized by the amount of linked documents that are analysed. However, we don't want the good quality original documents that have linked documents of poor relevance. Perhaps, the linked documents should have less influence on the score, when the original document scores well.

7.2.4 Multiple queries

While analysing the resulting documents from expanded queries, the problem of query drift was evident throughout. It is very hard to find a reliable query automatically that does not result in query drift occurring. One way to reduce the impact of query-drift is to extend the query expansion mechanism to make use of multiple expanded queries. Instead of outputting a single expanded query to the user, a new set of URLs could instead be outputted, where that set consisted of the relevant documents from the original query, and other relevant documents from the best expanded queries.

7.2.5 Additional vocabulary sources

The use of sub-queries to build vocabulary terms has proven to be an unreliable source. The performance of AbraQNew depends heavily on the vocabulary quality. It would be beneficial to employ more reliable sources. A good quality thesaurus was not tried in this project, and would be worth investigation to see if it can improve the vocabulary quality.

7.2.6 Expansion of multiple underrepresented aspects

AbraQNew is currently limited to expanding the most underrepresented aspect. One improvement would be to address two or more underrepresented aspects. A useful method would be to attempt to use any terms that intersect the vocabularies in the set of underrepresented aspects.

Bibliography

[1] Worldwidewebsize, "The size of the world wide web (worldwidewebsize.com)." WorldWideWebSize.com.

[2] A. Arampatzis and J. Kamps, "A study of query length," in *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 811–812, ACM, 2008.

[3] J. Xu and W. B. Croft, "Query expansion using local and global document analysis," in *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 4–11, ACM, 1996.

[4] D. W. Crabtree, P. Andreae, and X. Gao, "Exploiting underrepresented query aspects for automatic query expansion," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 191–200, ACM, 2007.

[5] L. Zighelnic and O. Kurland, "Query-drift prevention for robust query expansion," in *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 825–826, ACM, 2008.

[6] G. Salton, *The SMART Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1971.

[7] C.-H. Chang and C.-C. Hsu, "Integrating query expansion and conceptual relevance feedback for personalized web information retrieval," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1-7, pp. 621–623, 1998.

[8] J. Xu and W. B. Croft, "Improving the effectiveness of information retrieval with local context analysis," *ACM Trans. Inf. Syst.*, vol. 18, no. 1, pp. 79–112, 2000.

[9] O. Vechtomova and Y. Wang, "A study of the effect of term proximity on query expansion," *Journal of Information Science*, vol. 32, no. 4, pp. 324–333, 2006.

[10] G. A. Miller, "Wordnet - about wordnet." wordnet.princeton.edu.

[11] E. M. Voorhees, "Query expansion using lexical-semantic relations," in *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, (New York, NY, USA), pp. 61–69, Springer-Verlag New York, Inc., 1994.

[12] R. L. Cilibrasi and P. M. Vitanyi, "The google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 370–383, 2007.

[13] G. Inc, "Developer's guide - google ajax search api - google code." code.google.com.

[14] MIT, “Mit java wordnet interface (jwi).” projects.csail.mit.edu.

[15] M. Watson, “Fasttag 2.0: Part-of-speech tagger for java.” www.markwatson.com.

[16] U. o. T. Raymond J. Mooney, “Mit java wordnet interface (jwi).” www.cs.utexas.edu.

[17] TREC, “Trec 2005 hard track test topics.” http://trec.nist.gov.

[18] TREC, “Trec 2003 hard track test topics.” http://trec.nist.gov.