Query Expansion and Query Reduction in Document Retrieval*

Ingrid Zukerman
School of Computer Science
and Software Engineering
Monash University
Clayton, VICTORIA 3800
AUSTRALIA
ingrid@csse.monash.edu.au

Bhavani Raskutti
Telstra Research Laboratories
770 Blackburn Road
Clayton, VICTORIA 3168
AUSTRALIA
Bhavani.Raskutti@team.telstra.com

Yingying Wen
School of Computer Science
and Software Engineering
Monash University
Clayton, VICTORIA 3800
AUSTRALIA
ywen@csse.monash.edu.au

Abstract

We investigate two seemingly incompatible approaches for improving document retrieval performance in the context of question answering: query expansion and query reduction. Queries are expanded by generating lexical paraphrases. Syntactic, semantic and corpus-based frequency information is used in this process. Queries are reduced by removing words that may detract from retrieval performance. Features that identify these words were obtained from decision graphs. These approaches were evaluated using a subset of queries from TREC8, 9 and 10. Our evaluation shows that each approach in isolation improves retrieval performance, and both approaches together yield substantial improvements. Specifically, query expansion followed by reduction improved the average number of correct documents retrieved by 21.7% and the average number of queries that can be answered by 15%.

1. Introduction

One of the difficulties users face when searching for information in a knowledge repository is that of finding the words that will produce the desired outcome, e.g., relevant documents or precise answers. On one hand, the vocabulary users employ in their queries may be different from the vocabulary within particular Internet resources; on the other hand, users' vocabulary may not be discriminating enough. Both cases lead to retrieval failure.

In this paper, we investigate two seemingly incompatible approaches for improving document retrieval performance in the context of question answering: query expansion and query reduction.

We perform query expansion by generating lexical paraphrases of queries. These paraphrases replace content words in the queries with their synonyms. The following information sources are used in this process: syntactic information obtained using Brill's part-of-speech tagger [1]; semantic information obtained from WordNet [8] and the Webster-1913 on-line dictionary; and statistical information obtained from our document collection. The statistical information is used to moderate the alternatives obtained from the semantic resources, by preferring query paraphrases that contain frequent word combinations. A probabilistic formulation of the query paraphrases is then incorporated into the vector-space document-retrieval model [12].

Query reduction is performed by removing from queries words that may detract from retrieval performance. Attributes that identify these words were obtained by using decision graphs [10] to analyze the influence of different query attributes on retrieval performance. Three types of query attributes were considered: syntactic, paraphrase-based and frequency-based.

Our evaluation assessed the effect of paraphrase-based query expansion and of query reduction on document retrieval performance in the context of the TREC question-answering task. This task was selected since document retrieval is the first step in our project, whose eventual aim is to generate answers for users' queries. Our evaluation was performed on subsets of the TREC8, TREC9 and TREC10 collections. These subsets comprise queries whose answers reside in the LA Times portion of the TREC corpus (the other repositories were omitted owing to disk space limitations).

In the next section we describe related research. Section 3 discusses the query expansion process (resources, procedure and probabilistic formulation) and its evaluation. Section 4 describes the query reduction process (application of decision graphs) and its evaluation. In Section 5 we present concluding remarks.

* This research was supported in part by Australian Research Council grant DP0209565.

2. Related Research

The vocabulary mis-match between user queries and indexed documents is often addressed through *query expansion*. The problems due to query terms that are not sufficiently discriminating may be addressed by *query term weighting*.

Our research combines both of these approaches. Its query-expansion aspect is related to thesaurus-based query-expansion methods. These methods typically perform *word sense disambiguation (WSD)* prior to query expansion. Mihalcea and Moldovan [7] and Lytinen *et al.* [6] used WordNet [8] to obtain the sense of a word. In contrast, Schütze and Pedersen [13] and Lin [5] used a corpus-based approach where they automatically constructed a thesaurus on the basis of contextual information. The results obtained by Schütze and Pedersen and by Lytinen *et al.* are encouraging. However, experimental results reported in [3] indicate that the improvement in IR performance due to WSD is restricted to short queries, and that IR performance is very sensitive to disambiguation errors. Harabagiu *et al.* [4] offered a different form of query expansion, where they used WordNet to propose synonyms for the words in a query, and applied heuristics to select which words to paraphrase.

The query-expansion aspect of our work differs from traditional query expansion approaches in that our query expansion takes the form of alternative lexical paraphrases, each of which is assigned a weight that reflects corpus-based frequency information. Each of these paraphrases is then treated as a query during document retrieval.

The query-reduction aspect of our work is related to query-term weighting [11], which applies heuristics to reduce the weight of high-frequency query terms. In contrast, we use decision graphs to identify query-term attributes that detract from retrieval performance. Terms with these attributes are then removed from a copy of the original query and from paraphrases generated for this query.

Finally, this research is also related to Inference Nets [14], as the outcome of query expansion and reduction may be cast as terms in a query network.

3. Query Expansion

In this section, we discuss the resources used by our query paraphrasing mechanism, describe the paraphrasing process, and present a probabilistic formulation that incorporates query paraphrasing into the vector-space model. We then evaluate the retrieval performance of our mechanism.

3.1. Resources

Our system uses syntactic, semantic and statistical information for paraphrase generation.

Syntactic information for each query was obtained from Brill's part-of-speech (PoS) tagger [1].

Semantic information was obtained from two sources: WordNet – a knowledge-intensive, hand-built on-line repository; and Webster – an on-line version of the Webster-1913 dictionary (`http://www.dict.org`). WordNet was used to generate lemmas (uninflected versions of words) for the corpus and the queries, and to generate different types of synonyms for the words in the queries. Webster was used to automatically construct a list of nominals corresponding to the verbs in the corpus, and a list of verbs corresponding to the nouns in the corpus. The lemmas in these lists were used by WordNet to generate additional synonyms for the words in the queries. The idea was that nominalizations and verbalizations will help paraphrase queries such as "who *killed* Lincoln?" into "who is the *murderer* of Lincoln?" [4].[1]

The nominal list and the verb list were obtained by building a vector from the content lemmas in the definition of each word in the Webster dictionary, and applying the cosine measure to determine the similarity between the vector corresponding to each noun (or verb) in the dictionary and the vectors corresponding to the verbs (or nouns) in the dictionary. The verbs (or nouns) with the highest similarity measures to the original noun (or verb) and with the same stem were retained.

Statistical information was obtained from the LA Times portion of the NIST Text Research Collection (`http://trec.nist.gov`). This corpus, which was also used to test the retrieval performance of our system, was small enough to satisfy our disk space limitations, and sufficiently large to yield significant results (131,896 documents). Full-text indexing was performed for the documents in the LA Times collection using lemmas, rather than stems or complete words.

The statistical information was used to calculate the probability of the paraphrases generated for a query (Section 3.2.5). The statistical information was stored in a lemma dictionary (202,485 lemmas) and a lemma-pair dictionary (37,341,156 lemma pairs). Lemma pairs which appear only once constitute 64% of the pairs, and were omitted from our dictionary owing to disk space limitations.

3.2. Procedure

The following procedure is applied to paraphrase a query:

1. Tokenize, tag and lemmatize the query.
2. Generate replacement lemmas for each content lemma in the query.

1 It was necessary to build nominalization and verbalization lists because WordNet does not include this information.

3. Propose paraphrases for the query using different combinations of replacement lemmas, compute the probability of each paraphrase, and rank the paraphrases according to their probabilities. Retain the lemmatized query plus the top K paraphrases.

Documents are then retrieved for the query and its paraphrases, the probability of each document is calculated, and the top N documents are retained.

3.2.1. Tagging and lemmatizing the queries. We used Brill's tagger [1] to obtain the PoS of a word. This PoS is used to constrain the number of synonyms generated for a word. Brill's tagger incorrectly tagged 16% of the queries, which has a marginal detrimental effect on retrieval performance [16]. After tagging, each query was lemmatized (using WordNet).

3.2.2. Proposing replacements for each lemma. Two resources were used when proposing replacements for the content lemmas in a query: WordNet, and the nominalization and verbalization lists built from Webster. These resources were used as follows:

1. For each word in the query, we determined its lemma(s) and the lemma(s) that verbalize it (if it is a noun) or nominalize it (if it is a verb).
2. We then used WordNet to propose different types of synonyms for the lemmas produced in the first step. These types of synonyms were: `synonyms`, `attributes`, `pertainyms` and `seealsos` [8].[2] For example, according to WordNet, a `synonym` for "high" is "steep", an `attribute` is "height", and a `seealso` is "tall"; a `pertainym` for "chinese" is "China".

3.2.3. Paraphrasing queries. Query paraphrases were generated by an iterative process which considers each content lemma in a query in turn, and proposes a replacement lemma from those collected from our information sources (Section 3.2.2). Queries which do not have sufficient context are not paraphrased. These are queries where all the words except one are closed-class words or stop words (frequently occurring words that are ignored when used as search terms).

3.2.4. Probability of a paraphrase. The probability of a paraphrase depends on two factors: (1) how similar is the paraphrase to the original query, and (2) how common are the lemma combinations in the paraphrase. This may be expressed as follows:

$$\Pr(\text{Para}_i|\text{Query}) = \frac{\Pr(\text{Query}|\text{Para}_i) \times \Pr(\text{Para}_i)}{\Pr(\text{Query})} \quad (1)$$

where Para_i is the ith paraphrase of a query. Since the probability of the denominator is constant for a given query, we obtain:

$$\Pr(\text{Para}_i|\text{Query}) \propto \Pr(\text{Query}|\text{Para}_i) \times \Pr(\text{Para}_i) \quad (2)$$

where

$$\Pr(\text{Query}|\text{Para}_i) = \Pr(\text{Qlem}_1, \ldots, \text{Qlem}_L|\text{lem}_{i,1}, \ldots, \text{lem}_{i,L}) \quad (3)$$

$$\Pr(\text{Para}_i) = \Pr(\text{lem}_{i,1}, \ldots, \text{lem}_{i,L}) \quad (4)$$

where L is the number of content lemmas in a query, $\Pr(\text{Qlem}_j)$ is the probability of using Qlem_j – the jth lemma in the query, and $\Pr(\text{lem}_{i,j})$ is the probability of using $\text{lem}_{i,j}$ – the jth lemma in the ith paraphrase of the query.

To calculate $\Pr(\text{Query}|\text{Para}_i)$ in Eqn. 3 we assume
(1) $\Pr(\text{Qlem}_k|\text{lem}_{i,1}, \ldots, \text{lem}_{i,L})$ is independent of $\Pr(\text{Qlem}_j|\text{lem}_{i,1}, \ldots, \text{lem}_{i,L})$ for $k, j = 1, \ldots, L$ and $k \neq j$, and
(2) given $\text{lem}_{i,k}$, Qlem_k is independent of the other lemmas in the query paraphrase, i.e., $\Pr(\text{Qlem}_k|\text{lem}_{i,1}, \ldots, \text{lem}_{i,L}) \cong \Pr(\text{Qlem}_k|\text{lem}_{i,k})$.

These assumptions yield

$$\Pr(\text{Query}|\text{Para}_i) \cong \prod_{j=1}^{L} \Pr(\text{Qlem}_j|\text{lem}_{i,j}) \quad (5)$$

Eqn. 4 may be rewritten using Bayes rule:

$$\Pr(\text{Para}_i) \cong \prod_{j=1}^{L} \Pr(\text{lem}_{i,j}|\text{ctxt}_{i,j}) \quad (6)$$

where $\text{ctxt}_{i,j}$ is the context for lemma j in the ith paraphrase.

Substituting Eqn. 5 and Eqn. 6 into Eqn. 2 yields

$$\Pr(\text{Para}_i|\text{Query}) \propto \prod_{j=1}^{L} \left[\Pr(\text{Qlem}_j|\text{lem}_{i,j}) \times \Pr(\text{lem}_{i,j}|\text{ctxt}_{i,j})\right] \quad (7)$$

$\boldsymbol{Pr}(\boldsymbol{Qlem}_j|\boldsymbol{lem}_{i,j})$ may be interpreted as the probability of using Qlem_j instead of $\text{lem}_{i,j}$. Intuitively, this probability depends on the similarity between the lemmas. At present, we use the baseline similarity measure $\Pr(\text{Qlem}_j|\text{lem}_{i,j}) = 1$ if $\text{lem}_{i,j}$ is a WordNet synonym of Qlem_j (where "synonym" encompasses different types of WordNet similarities). We are currently considering some of the WordNet similarity measures described in [2].

2 In preliminary experiments we also generated `hypernyms` and `hyponyms`. However, this increased exponentially the number of alternative paraphrases, without improving retrieval performance. Also, in previous experiments we considered alternative semantic resources, but the best results were obtained with WordNet.

$\boldsymbol{Pr}(\boldsymbol{lem}_{i,j}|\boldsymbol{ctxt}_{i,j})$ may be represented by $\Pr(\text{lem}_{i,j}|\text{lem}_{i,1},\ldots,\text{lem}_{i,j-1})$, which we approximate as follows:

$$\Pr(\text{lem}_{i,j}|\text{lem}_{i,1},\ldots,\text{lem}_{i,j-1}) \simeq \prod_{k=1}^{j-1} \Pr(\text{lem}_{i,k} \triangleright \text{lem}_{i,j}) \quad (8)$$

where $\Pr(\text{lem}_{i,k} \triangleright \text{lem}_{i,j})$ – the probability that lemma k in the ith paraphrase is followed by lemma j – is obtained directly from the lemma-pair dictionary (Section 3.1).

This approximation, although *ad hoc*, works well in practice, yielding a better performance than bi-gram approximations [16].

3.2.5. Retrieving documents for each query. Our retrieval procedure incorporates query paraphrases into the vector-space model, which calculates the score of candidate documents given a list of terms in a query. Normally, this score is based on the TF.IDF measure, which for the ith paraphrase of a query yields the following formula:

$$\textit{Score}(\text{Doc}|\text{Para}_i) = \sum_{j=1}^{L} \textit{tfidf}(\text{Doc},\text{lem}_{i,j}) \quad (9)$$

By normalizing the scores of the documents, we obtain a probability that a document contains the answer to the ith paraphrase:

$$\Pr(\text{Doc}|\text{Para}_i) \propto \sum_{j=1}^{L} \textit{tfidf}(\text{Doc},\text{lem}_{i,j}) \quad (10)$$

Let us now consider different paraphrases of a query, and assume that, given a paraphrase, a document retrieved on the basis of the paraphrase is conditionally independent of the original query. This yields the following formula:

$$\Pr(\text{Doc}|\text{Query}) = \sum_{i=0}^{n} \Pr(\text{Doc}|\text{Para}_i) \times \Pr(\text{Para}_i|\text{Query}) \quad (11)$$

where n is the number of paraphrases. We also adopt the convention that the 0-th paraphrase is the original lemmatized query.

By substituting Eqn. 10 and Eqn. 7 for the first and second factors in Eqn. 11 respectively we obtain

$$\Pr(\text{Doc}|\text{Query}) = \sum_{i=0}^{n} \left[\sum_{j=1}^{L} \textit{tfidf}(\text{Doc},\text{lem}_{i,j}) \right] \times \left[\prod_{j=1}^{L} \left[\Pr(\text{Qlem}_j|\text{lem}_{i,j}) \times \Pr(\text{lem}_{i,j}|\text{ctxt}_{i,j}) \right] \right] \quad (12)$$

3.3. Evaluation

In this section we describe the metrics used to evaluate the retrieval performance of our system, discuss our evaluation experiment, and analyze our results.

3.3.1. Evaluation metrics. We employ two measures of retrieval performance: (1) *total correct documents*, which returns the number of correct documents retrieved for all the queries (this measure is similar, but not equivalent, to the standard recall measure); and (2) *number of answerable queries*, which returns the number of queries for which the system has retrieved at least one document that contains the answer to the query.

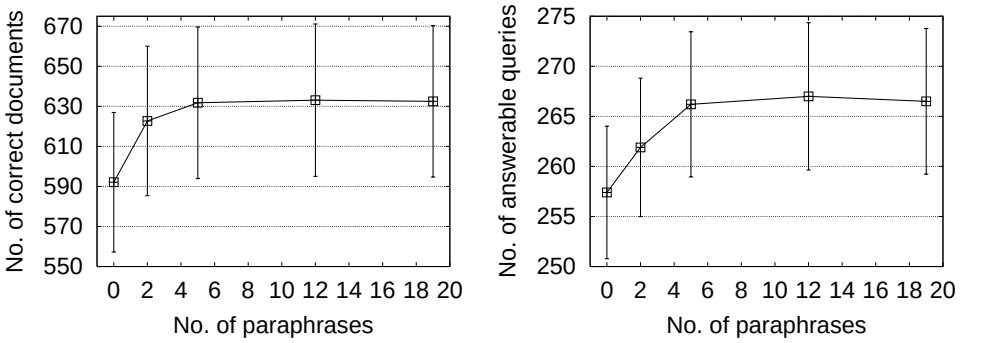
These measures were chosen for the following reasons. In the question-answering task we want to maximize the chances of finding the answer to a user's query. The hope is that returning a large number of documents that contain this answer (measured by *total correct documents*) will be helpful during the answer extraction phase of this project. However, this measure alone is not sufficient to evaluate the performance of our system, as even with a high number of correct documents, it is possible that we are retrieving many correct documents for relatively few queries, leaving many queries unanswered.

The standard precision measure, commonly used in retrieval tasks, does not address this problem. For instance, consider a situation where 10 correct documents are retrieved for each of 2 queries and 0 correct documents for each of 3 queries, compared to a situation where 2 correct documents are retrieved for each of 5 queries. Average precision would yield a better score for the first situation, failing to address the question of interest for the question-answering task, namely how many queries have a chance of being answered, which is 2 in the first case and 5 in the second case. This is the number represented in our second measure of performance – *number of answerable queries*.

3.3.2. Experiment. Our evaluation determines the effect of paraphrase-based query expansion on retrieval performance, as well as the number of paraphrases that yields the best performance. The number of retrieved documents is kept constant at 200, as suggested in [9].[3]

For each run, we submitted to the retrieval engine increasing sets of paraphrases as follows: first the lemmatized query alone (Set 0), next the query plus up to 2 paraphrases (Set 2), then the query plus up to 5 paraphrases (Set 5), the query plus up to 12 paraphrases (Set 12) and the query plus a maximum of 19 paraphrases (Set 19). These numbers represent the *maximum* number of paraphrases for a query –

3 In a related experiment, we varied the number of retrieved documents while keeping the number of paraphrases constant. This experiment showed that query paraphrasing reduces the number of documents that need to be retrieved to achieve a particular level of performance.



(a) Average number of correct documents (b) Average number of answerable queries

Figure 1. Effect of number of paraphrases on retrieval performance for 380 TREC queries (10 random samples, 200 retrieved documents).

fewer paraphrases are generated if there aren't enough synonyms.[4]

We ran the query expansion process on 10 random samples of 380 queries each. These samples were extracted from the 760 TREC8, TREC9 and TREC10 queries whose answers appear in the LA Times portion of the TREC document collection.[5] The average retrieval performance obtained for these 10 samples is depicted in Figure 1; the error bars represent 1 standard deviation. Figure 1(a) depicts the average number of correct documents retrieved as a function of the number of paraphrases generated, and Figure 1(b) shows the average number of answerable queries. To put these plots in perspective, of the 131,896 documents in the LA Times repository, 2239 documents were judged correct for 760 of the 1393 TREC queries. Further, the maximum number of correct documents varies for each random sample (averaging at 1097.5), while the maximum number of answerable queries remains constant at 380.

We obtain from Figure 1 that query paraphrasing yields an average improvement of 6.8% in the number of correct documents, and an average improvement of 3.5% in the number of answerable queries. That is, *query paraphrasing yields a modest increase in the number of correct documents retrieved and in the number of answerable queries.*

It is worth noting that these improvements are due to both the lemmas that were paraphrased and those that were *not* paraphrased. Paraphrasing important lemmas adds words to a query which hopefully match the language in the target documents. In contrast, paraphrasing non-essential lemmas leaves the important lemmas untouched (and repeated) in many paraphrases, which effectively increases their relative weight in the retrieval process.

3.3.3. Retrieval performance: three collections. The retrieval performance of query paraphrasing was also evaluated separately for each of the three TREC query collections. Table 1 summarizes this retrieval performance compared with the baseline performance without expansion. The first four columns contain: (1) the name of the collection, (2) the total number of queries available for the collection, (3) the number of queries that have answers in the LA Times subset of the TREC document collection, and (4) the number of documents that contain answers for the queries in each collection. For instance, from a total of 131,896 documents in the LA Times subset, there were 480 documents which were judged correct for 125 of the 200 TREC8 queries. The next two columns show the total correct documents and answerable queries without query expansion, and the last two columns show these metrics with paraphrase-based expansion. The best improvements (obtained with WordNet for TREC9) are boldfaced.

The results in Table 1 show that the performance improvements obtained from paraphrase-based query expansion are marginal for TREC8 and TREC10, but are more substantial for TREC9. Further, these results show that there are significant differences in baseline performance for the three collections.

4. Query Reduction

The differences in retrieval performance for the three TREC collections prompted us to study the problem of using observable features of queries to predict retrieval performance. We used as our analysis tool decision graphs [10]

4 Previous experiments with increasing numbers of paraphrases show that Sets 0, 2, 5, 12 and 19 are significant in terms of retrieval performance. Also, experiments with up to 40 paraphrases show that there is no advantage in generating more than 19 paraphrases.

5 Randomized samples are not necessary to evaluate the query expansion process. However, we used such samples to obtain a baseline performance measure against which we can compare the results obtained from query reduction (Section 4).

Collection	# Total queries	# LA Times queries	# docs judged correct	Baseline (no expansion)		Expansion (WordNet)	
				# Correct docs	# Answerable queries	# Correct docs	# Answerable queries
TREC8	200	125	480	242 (50.4%)	90 (72.0%)	254 (52.9%)	92 (73.6%)
TREC9	693	404	1232	596 (48.4%)	251 (62.0%)	**663 (53.8%)**	**268 (66.0%)**
TREC10	500	231	527	350 (66.4%)	171 (74.0%)	359 (68.1%)	174 (75.3%)
Total	1393	760	2239	1188 (53%)	512 (67.4%)	1276 (57%)	534 (70.3%)

Table 1. Summary of retrieval performance for TREC8, TREC9 and TREC10.

– an extension of the decision trees described in [15]. In this section we describe the query features considered in the decision-graph analysis, and present the insights obtained from this analysis. We then discuss the incorporation of these insights into our document retrieval process, and present the results of our evaluation.

4.1. Decision-graph analysis

Decision graphs (and decision trees) determine which of a set of attributes may be used to predict membership in a class of interest. In our case, this class is "answerable query". The input to *Dgraf* – the decision-graph program – consisted of the class membership of each query plus 28 query attributes. These attributes belong to three categories: *syntactic* – 9 attributes of the query itself, such as query length and number of nouns; *paraphrase-based* – 1 attribute – number of paraphrases; and *frequency-based* – 18 corpus-based attributes of the query, such as frequency of the nouns, verbs and proper nouns in the query.

Dgraf was trained on 10 random samples of 380 queries (and their paraphrases) extracted from the 760 queries whose answers appear in the LA Times portion of the TREC collection. The holdout sets for these random samples correspond to the 10 query sets used to evaluate the query-expansion process. All the runs yielded two query attributes that together are good predictors of retrieval performance: (1) *noun frequency*, and (2) *proper noun frequency*. That is, for each run, Dgraf split on both of these attributes, yielding a decision graph containing a leaf that is characterized as follows:

$$\textit{noun frequency} < \mathrm{Thr}_{Noun} \text{ and}$$
$$\textit{proper noun frequency} < Thr_{PropNoun}$$

This leaf defines a region of high retrieval accuracy. Specifically, averaging over the 10 Dgraf runs, 91.3% of the queries in this leaf were answerable by the retrieved documents. It is worth noting that although all the runs identified the same general attributes, they did not produce the same thresholds. For instance, Thr_{Noun} was 1519 for Sample 2 and 755 for Sample 9.

4.2. Implementation of the decision-graph results

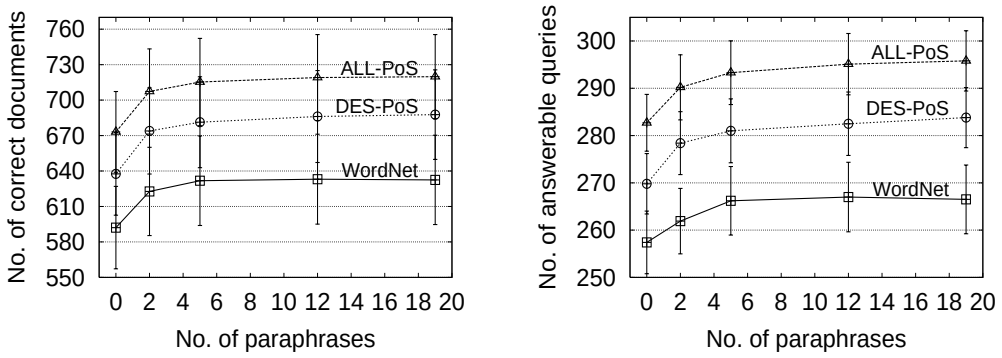
The results obtained by Dgraf were implemented as a rule that was applied in a post-processing step of the query paraphrasing process (i.e., Step 4 in the procedure described in Section 3.2). We considered two ways of applying this rule: *Designated-PoS* and *All-PoS*. The Designated-PoS policy removes only the lemmas whose PoS was identified by Dgraf (i.e., nouns and proper nouns) and whose frequency exceeds the threshold determined by Dgraf. In contrast, the All-PoS policy extends the results obtained by Dgraf to remove lemmas with other PoS (verbs, adjectives and adverbs) if their frequency exceeds the Dgraf threshold for nouns. The resulting post-processing rules are:

Designated-PoS. *Remove all the nouns whose frequency is greater than* Thr_{Noun} *and all the proper nouns whose frequency is greater than* $Thr_{PropNoun}$.

All-PoS. *Remove all the nouns, verbs, adjectives and adverbs whose frequency is greater than* Thr_{Noun} *and all the proper nouns whose frequency is greater than* $Thr_{PropNoun}$.

Both rules reflect the observation that high-frequency lemmas may lead the retrieval process astray, and that performance may be improved by removing these lemmas. For instance, consider the query "Where does Mother Angelica live?". There are 22,957 documents that contain the lemma "live", 7910 documents that contain "mother", and 59 documents that contain "angelica". In this case, the retrieval process may return many documents that contain only "mother" and "live", leaving documents containing "angelica" out of the top-200 retrieved documents. The expectation from these rules is that retrieval performance will be improved by removing "live", "mother" or both.

These rules were applied to both the lemmatized query and its paraphrases. However, if the frequency of all the content lemmas in a query (or its paraphrase) exceeded the threshold for the corresponding PoS, the lemma with the smallest threshold violation was retained (this is the lemma with the lowest frequency-to-threshold ratio). In addition, two copies of the lemmatized query were retained: the original and a "reduced" copy (after the application of the reduction rule).



(a) Average number of correct documents (b) Average number of answerable queries

Figure 2. Effect of query reduction and number of paraphrases on retrieval performance for 380 TREC queries (10 random samples, 200 retrieved documents).

4.3. Evaluation

Our two query-reduction rules were evaluated using the holdout sets for the 10 random query sets used to train Dgraf (Section 4.1). As stated above, these holdout sets were also used to evaluate the query-expansion process. The average retrieval performance obtained for these 10 samples is depicted in Figure 2; the error bars represent 1 standard deviation (the results obtained using WordNet for query expansion are included for comparison purposes). Figure 2(a) depicts the average number of correct documents retrieved as a function of the number of paraphrases generated, and Figure 2(b) shows the average number of answerable queries. The results for 0 paraphrases depict retrieval performance for query reduction alone. The results for 2, 5, 12 and 19 paraphrases depict the effect of query expansion followed by reduction.

As seen in Figure 2, the All-PoS policy produced the best results, significantly improving retrieval performance both with and without query expansion. We postulate that All-PoS performs better than Designated-PoS, because Designated-PoS leaves in the query some high-frequency content lemmas that may still lead the retrieval process astray, while All-PoS removes all such lemmas. It is also worth noting that the improvement obtained with query reduction alone exceeds that obtained with query expansion alone, and that the improvement obtained by applying query expansion followed by reduction is larger than the sum of the improvements obtained using expansion alone and reduction alone. We posit that this happens when query expansion replaces non-essential lemmas with their synonyms, yielding paraphrases where the essential lemmas are repeated (Section 3). Query reduction then removes those synonyms that have a high frequency, yielding an even heavier relative weighting for the essential lemmas.

Our results also indicate that, with the exception of very short queries (2 or 3 words), the improvements obtained from query expansion and query reduction seem independent of query length. Query expansion had a modest positive effect for most query lengths, query reduction had a substantial positive effect, and expansion followed by reduction generally outperformed each method in isolation.

Table 2 summarizes the main results from Figure 2 according to the query-processing method: baseline, paraphrase-based query expansion only (WordNet), query reduction only (All-PoS), and expansion followed by reduction (WordNet+ All-PoS). The second and fifth columns contain the average number of retrieved correct documents and the average number of answerable queries respectively. The third and sixth columns show the average improvement obtained by each of the three expansion/reduction methods compared to the baseline performance for correct documents and answerable queries respectively. Finally, the fourth and last columns show an additional performance measure, which we call *improvement of method$_i$ compared to the maximum possible improvement*. This measure, which is expressed by the following formula, reflects how much of the "slack" (room for improvement) left by the baseline method has been picked up by method$_i$.

$$\frac{\text{performance-of-method}_i - \text{baseline-performance}}{\text{maximum-possible-performance} - \text{baseline-performance}}$$

The results from Figure 2 and Table 2 show that *query reduction yields a significant increase in the number of correct documents retrieved and in the number of answerable queries, and that query expansion followed by reduction yields even more substantial improvements.*

Method	Average Correct Docs	Average Improv	Average Improv Comp Max	Average Answerable Queries	Average Improv	Average Improv Comp Max
Baseline	592.1	–	–	257.5	–	–
WordNet	632.5	6.8	8.3	266.5	3.5	7.4
All-PoS	673.1	13.7	16.4	282.7	9.9	20.6
All-PoS+WordNet	719.8	21.7	25.8	295.8	15	31.3

Table 2. Comparison of retrieval performance for query expansion and reduction methods.

5. Conclusion

We have investigated the effect of paraphrase-based query expansion and of query reduction on document retrieval performance. Query expansion was performed using syntactic, semantic and statistical information. Query reduction was performed by applying rules that implement insights obtained from decision graphs. Our results show that: (a) paraphrase-based query expansion yields a modest improvement in document retrieval performance; (b) analysis based on decision graphs yields factors that influence retrieval performance; (c) query reduction based on these factors significantly improves retrieval performance; and (d) query expansion followed by reduction yields even more substantial improvements in retrieval performance.

References

[1] E. Brill. A simple rule-based part of speech tagger. In *ANLP-92 – Proceedings of the Third Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.

[2] A. Budanitsky and G. Hirst. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania, 2000.

[3] J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. Indexing with WordNet synsets can improve text retrieval. In *Proceedings of the COLING-ACL'98 Workshop on Usage of WordNet in Natural Language Processing Systems*, pages 38–44, Montreal, Canada, 1998.

[4] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. The role of lexico-semantic feedback in open domain textual question-answering. In *ACL01 – Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 274–281, Toulouse, France, 2001.

[5] D. Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL'98 – Proceedings of the International Conference on Computational Linguistics and the Annual Meeting of the Association for Computational Linguistics*, pages 768–774, Montreal, Canada, 1998.

[6] S. Lytinen, N. Tomuro, and T. Repede. The use of WordNet sense tagging in FAQfinder. In *Proceedings of the AAAI00 Workshop on AI and Web Search*, Austin, Texas, 2000.

[7] R. Mihalcea and D. Moldovan. A method for word sense disambiguation of unrestricted text. In *ACL99 – Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, Baltimore, Maryland, 1999.

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

[9] D. Moldovan, M. Pasca, S. Harabagiu, and M. Surdeanu. Performance issues and error analysis in an open domain question answering system. In *ACL02 – Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, Philadelphia, Pennsylvania, 2002.

[10] J. J. Oliver. Decision graphs – an extension of decision trees. In *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics*, pages 343–350, Fort Lauderdale, Florida, 1993.

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

[12] G. Salton and M. McGill. *An Introduction to Modern Information Retrieval*. McGraw Hill, 1983.

[13] H. Sch¨utze and J. O. Pedersen. Information retrieval based on word senses. In *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, Nevada, 1995.

[14] H. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, 1991.

[15] C. Wallace and J. Patrick. Coding decision trees. *Machine Learning*, 11:7–22, 1993.

[16] I. Zukerman and B. Raskutti. Lexical query paraphrasing for document retrieval. In *COLING'02 – Proceedings of the International Conference on Computational Linguistics*, pages 1177–1183, Taipei, Taiwan, 2002.