Lexical Query Paraphrasing for Document Retrieval*

Ingrid Zukerman
School of Computer Science and Software Eng.
Monash University
Clayton, VICTORIA 3800
AUSTRALIA

Bhavani Raskutti
Telstra Research Laboratories
770 Blackburn Road
Clayton, VICTORIA 3168
AUSTRALIA

Abstract

We describe a mechanism for the generation of lexical paraphrases of queries posed to an Internet resource. These paraphrases are generated using WordNet and part-of-speech information to propose synonyms for the content words in the queries. Statistical information, obtained from a corpus, is then used to rank the paraphrases. We evaluated our mechanism using 404 queries whose answers reside in the LA Times subset of the TREC-9 corpus. There was a 14% improvement in performance when paraphrases were used for document retrieval.

1 Introduction

The vocabulary of users of domain-specific retrieval systems often differs from the vocabulary within a particular resource, leading to retrieval failure. In this research, we address this problem by submitting multiple paraphrases of a query to a retrieval system, in the hope that one or more of the posited paraphrases will match a relevant document.

We focus on the generation of lexical paraphrases for queries posed to the Internet. These are paraphrases where content words are replaced with synonyms. We use WordNet (Miller et al., 1990) and part-of-speech information to propose these synonyms, and build candidate paraphrases from combinations of these synonyms. The resultant paraphrases are then scored using word co-occurrence information obtained from a corpus, and the highest scoring paraphrases are retained. Our evaluation shows a 14% improvement in retrieval performance as a result of query paraphrasing.

In the next section we describe related research. In Section 3, we discuss the resources used by our mechanism. The paraphrase generation and document retrieval processes are described in Section 4. Section 5 presents sample paraphrases, followed by our evaluation and 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. Two common techniques for query expansion are *blind relevance feedback* (Buckley et al., 1995; Mitra et al., 1998) and *word sense disambiguation (WSD)* (Mihalcea and Moldovan, 1999; Lytinen et al., 2000; Schütze and Pedersen, 1995; Lin, 1998). Blind relevance feedback consists of retrieving a small number of documents using a query given by a user, and then constructing an expanded query that includes content words that appear frequently in these documents. This expanded query is used to retrieve a new set of documents. WSD often precedes query expansion to avoid retrieving irrelevant information. Mihalcea and Moldovan (1999) and Lytinen *et al.* (2000) used a machine readable thesaurus, specifically WordNet (Miller et al., 1990), to obtain the sense of a word, while Schütze and Pedersen (1995) and Lin (1998) used automatically constructed thesauri.

The improvements in retrieval performance reported in (Mitra et al., 1998) are comparable to those reported here (note that these researchers consider precision, while we consider recall). The results obtained by Schütze and Pedersen (1995) and by Lytinen *et al.* (2000) are encouraging. However, experimental results reported in (Sanderson, 1994; Gonzalo et al., 1998) 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.

Our approach to document retrieval differs from the above approaches in that the expansion of a query takes the form of alternative lexical paraphrases. Like Harabagiu *et al.* (2001), we use WordNet to propose synonyms for the words in a query. However, they apply heuristics to select which words to paraphrase. In contrast, we use corpus-based information in the context of the entire query to calculate the score of a paraphrase

and select which paraphrases to retain, and then use the paraphrase scores to influence the document retrieval process.

3 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 (Brill, 1992). Semantic information consisting of different types of synonyms for the words in each query was obtained from WordNet (Miller et al., 1990).

The corpus used for information retrieval and for the collection of statistical information was the LA Times portion of the NIST Text Research Collection (`//trec.nist.gov`). This corpus was small enough to satisfy our disk space limitations, and sufficiently large to yield statistically significant results (131,896 documents). Full-text indexing was performed for the documents in the LA Times collection, using lemmas (rather than words). The lemmas for the words in the LA Times collection were also obtained from WordNet (Miller et al., 1990).

The statistical information was used to assign a score to the paraphrases generated for a query (Section 4.4). This information was stored in a lemma dictionary (202,485 lemmas) and a lemma-pair dictionary (37,341,156 lemma-pairs). The lemma dictionary associates with each lemma the number of times it appears in the corpus. The lemma-pair dictionary associates with each ordered lemma-pair l_i-l_j the number of times l_i appears before l_j in a five-word window in the corpus (not counting stop words and closed-class words). The dictionary maintains a different entry for the lemma pair l_j-l_i. Lemma-pairs which appear only once constitute 64% of the pairs, and were omitted from our dictionary owing to disk space limitations.

4 Paraphrasing and Retrieval Procedure

The procedure for paraphrasing a query consists of the following steps:

1. Tokenize, tag and lemmatize the query.
2. Generate synonyms for each content lemma in the query (stop words are ignored).
3. Propose paraphrases for the query using different synonym combinations, compute a score for each paraphrase, and rank the paraphrases according to their score. The lemmatized query plus the 19 top paraphrases are retained.

Documents are then retrieved for the query and its paraphrases.

4.1 Tagging and lemmatizing the queries

We used the part-of-speech (PoS) of a word to constrain the number of synonyms generated for it. Brill's tagger correctly tagged 84% of the queries. In order to determine the effect of tagging errors on retrieval performance, we corrected manually the wrong tags, and ran our system with both automatically-obtained and manually-corrected tags (Section 6). After tagging, each query was lemmatized (using WordNet). This was done since the index used for document retrieval is lemma-based.

4.2 Proposing synonyms for each word

The following types of WordNet synonyms were generated for each content lemma in a query: `synonyms`, `attributes`, `pertainyms` and `seealsos` (Miller et al., 1990).[1] 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". In order to curb the combinatorial explosion, we do not allow multiple-word synonyms for a lemma, and do not generate synonyms for proper nouns or stop words.

4.3 Paraphrasing queries

Query paraphrases are generated by an iterative process which considers each content lemma in a query in turn, and proposes a synonym from those collected from WordNet (Section 4.2). Queries which do not have sufficient context are not paraphrased. These are queries where all the words except one are stop words or closed-class words.

4.4 Computing paraphrase scores

The score of a paraphrase is based on how common are the lemma combinations in it. Ideally, this score would be represented by $\Pr(l_1, \ldots, l_n)$, where n is the number of lemmas in the paraphrase. However, in the absence of sufficient information to compute this joint probability, approximations based on conditional probabilities are often used, e.g.,

$$\Pr(l_1, \ldots, l_n) \simeq \Pr(l_n|l_{n-1}) \times \ldots \times \Pr(l_2|l_1) \times \Pr(l_1)$$

Unfortunately, this approximation yielded poor paraphrases in preliminary trials. We postulate that this is due to two reasons: (1) it takes into account the interaction between a lemma l_i and only one other lemma (without considering the rest of the lemmas in the query), and (2) relatively infrequent lemma combinations involving one frequent lemma

[1] In preliminary experiments we also generated `hypernyms` and `hyponyms`. However, this increased the number of alternative paraphrases exponentially, without improving the quality of the results in most cases.

are penalized (which is correct for conditional probabilities). For instance, if l_j appears 10 times in the corpus and l_i-l_j appears 4 times, $P(l_i|l_j) = 0.4\alpha$ (where α is a normalizing constant). In contrast, if l'_j appears 200 times in the corpus and l'_i-l'_j appears 30 times, $P(l'_i|l'_j) = 0.15\alpha$. However, l'_i-l'_j is a more frequent lemma combination, and should contribute a higher score to the paraphrase.

To address these problems, we propose using the joint probability of a pair of lemmas instead of their conditional probability. In the above example, this yields $P(l_i, l_j) = 4\beta$ and $P(l'_i, l'_j) = 30\beta$ (where β is a normalizing constant). These probabilities reflect more accurately the goodness of paraphrases containing these lemma-pairs. The resulting approximation of the probability of a paraphrase composed of lemmas $l_1, \ldots, l_n$ is as follows:

$$\Pr(l_1, \ldots, l_n) \simeq \prod_{i=1}^{n} \prod_{j=i+1}^{n} \Pr(l_i, l_j) \qquad (1)$$

$\Pr(l_i, l_j)$ is obtained directly from the lemma-pair frequencies, yielding

$$\Pr(l_1, \ldots, l_n) \simeq \prod_{i=1}^{n} \prod_{j=i+1}^{n} \beta \times \text{freq}(l_i, l_j)$$

where β is a normalizing constant.[2] Since this constant is not informative with respect to the relative scores of the paraphrases for a particular query, we drop it from consideration, and use only the frequencies to calculate the score of a paraphrase. Thus, our paraphrase scoring function is

$$\mathcal{PS}(l_1, \ldots, l_n) = \prod_{i=1}^{n} \prod_{j=i+1}^{n} \text{freq}(l_i, l_j) \qquad (2)$$

4.4.1 Experimental parameters

When calculating the score of a paraphrase using Equation 2, the following aspects regarding $\text{freq}(l_i, l_j)$ must be specified: (1) the extent to which the order of l_i and l_j (as it appears in the paraphrase) should be enforced; and (2) how to handle l_i-l_j pairs in the paraphrase that are absent from the lemma-pair dictionary. To illustrate these aspects, consider the candidate paraphrase "who is the greek deity of the ocean?" (proposed for "who is the greek god of the sea?"). The first aspect determines whether the frequency of only "greek deity" should be used, or whether "deity greek" should also be taken into account. The second aspect determines how to score the paraphrase if "greek ocean" is absent from the lemma-pair dictionary. These aspects are specified as experimental parameters of the system.

[2] $\beta = \frac{1}{\#\text{ of lemma-pairs}^{n(n-1)/2}} = \frac{1}{37{,}341{,}156^{n(n-1)/2}}$.

Relative word order. The extent to which we enforce the order of l_i-l_j when calculating $\text{freq}(l_i, l_j)$ is determined by the weight W_{order} as follows:

$$\text{freq}(l_i, l_j) = \text{freq}(l_i \rightarrow l_j) + W_{order} \times \text{freq}(l_j \rightarrow l_i) \qquad (3)$$

where $\text{freq}(l_i \rightarrow l_j)$ is the frequency of the lemma-pair (l_i, l_j) when l_i is followed by l_j. $W_{order} = 0$ allows only the word order in the paraphrase, while $W_{order} = 1$ counts equally the order in the paraphrase and the reverse order. We experimented with weights of 0, 1 and 0.5 for W_{order} (Section 6).

Absent lemma-pairs. When a lemma-pair is not in the dictionary, a frequency of 0 is returned. Using this frequency is too strict, because it invalidates an entire paraphrase on account of one culprit which may actually be innocent (recall that 64% of the lemma-pairs in the corpus – approximately 66 million pairs – had a frequency of 1 but were not stored). To address this problem, we assigned a penalty frequency of *AbsFreq* = 0.1 to a lemma-pair in a paraphrase that does not appear in the dictionary. That is, the score of a paraphrase is divided by 10 for each of its lemma-pairs that is absent from the dictionary.

In addition, we defined the experimental parameter *AbsAdjDiv*, which models the impact of adjacent lemma-pairs on paraphrasing and retrieval performance. This parameter takes the form of a divisor for *AbsFreq*: it stipulates by how much to divide *AbsFreq* for a lemma-pair that is adjacent in the paraphrase but absent from the dictionary. In the above example, *AbsAdjDiv*=10 would cause an absent "deity ocean" to receive a penalty of 0.01 (=0.1/10) compared to an absent "greek ocean", which would receive a penalty of 0.1. We experimented with four values for *AbsAdjDiv*: 1, 2, 10 and 20 (Section 6).

4.5 Retrieving documents for each query

Our retrieval process differs from the standard one in that for each query Q, we adjust the scores of the retrieved documents according to the scores of the paraphrases of Q (obtained from Equation 2). Our retrieval process consists of the following steps:

1. For each paraphrase P_i of Q ($i = 0, \ldots,$ #_para_Q), where P_0 is the lemmatized query:
   (a) Extract the content lemmas from P_i: $l_{i,1}, \ldots l_{i,N}$, where N is the number of content lemmas in paraphrase P_i.
   (b) For each lemma, compute a score for the retrieved documents using a standard IR measure, e.g., Term Frequency Inverse Document Frequency (TFIDF) (Salton and McGill, 1983). Let $tfidf(D_k, l_{i,j})$ be the score of

document D_k retrieved for lemma $l_{i,j}$ ($j = 1, \ldots, N$). When a document D_k is retrieved by more than one lemma in a paraphrase P_i, its TFIDF scores are added, yielding the score $\sum_{j=1}^{N} tfidf(D_k, l_{i,j})$. This score indicates how well D_k matches the lemmas in paraphrase P_i. In order to take into account the plausibility of P_i, this score is multiplied by $\mathcal{PS}(P_i)$ – the score of P_i obtained from Equation 2. This yields $\mathcal{DS}_{k,i}$, the score of document D_k for paraphrase P_i.

$$\mathcal{DS}_{k,i} = \mathcal{PS}(P_i) \times \sum_{j=1}^{N} tfidf(D_k, l_{i,j}) \quad (4)$$

2. For each document D_k, add the scores from each paraphrase (Equation 4), yielding

$$\mathcal{DS}_k = \sum_{i=1}^{\#\_para\_Q} \mathcal{PS}(P_i) \times \sum_{j=1}^{N} tfidf(D_k, l_{i,j}) \quad (5)$$

An outcome of this method is that lemmas which appear in several paraphrases receive a higher weight. This indirectly identifies the important words in a query, which positively affects retrieval performance (Section 6).

5 Sample Results

Table 1 shows the top 10 paraphrases generated by our system for three sample queries, and the 7 paraphrases generated for a fourth query (the lemmatized query is listed first). These paraphrases were obtained with $W_{order} = 1$, *AbsAdjDiv* = 10, and manually-corrected tagging (Section 4). The third column contains the paraphrase, the first column contains its score, and the second column contains the number of lemma-pairs in the paraphrase which were not found in the dictionary.

These examples illustrate the combined effect of contextual information and WordNet senses. The first query yields mostly felicitous paraphrases, despite their low overall score and absent lemma-pairs. This outcome may be attributed to the generally appropriate synonyms returned by WordNet for the lemmas in this query. The second query produces a mixed paraphrasing performance. The problematic paraphrases are generated because our corpus-based information supports WordNet's inappropriate suggestions of "manufacture" as a synonym for "invent" and "video" as a synonym for "television", thus yielding highly-ranked but incorrect paraphrases. The third query is an extreme example of this behaviour, where WordNet synonyms conspire with contextual information to steer the paraphrasing process toward inappropriate synonyms of "bear". The final example illustrates the opposite case, where the corpus information overcomes the effect of WordNet's less appropriate suggestions, which yield low-scoring paraphrases.

Table 1: Sample query paraphrases

Score	#Abs	Paraphrase
Who is the Greek God of the Sea ?		
9.20E+02	0	who be the greek god of the sea ?
6.90E+00	1	who be the greek god of the ocean ?
5.00E-01	1	who be the greece god of the sea ?
1.00E-02	2	who be the greece deity of the sea ?
1.00E-02	2	who be the greece divinity of the sea ?
1.00E-02	2	who be the greece immortal of the sea ?
1.00E-02	2	who be the greece idol of the sea ?
8.00E-03	2	who be the greek deity of the sea ?
8.00E-03	2	who be the greek divinity of the sea ?
8.00E-03	2	who be the greek immortal of the sea ?
8.00E-03	2	who be the greek idol of the sea ?
Who invented television ?		
7.00E+00	0	who invent television ?
1.60E+01	0	who manufacture television ?
1.60E+01	0	who manufacture video ?
1.10E+01	0	who manufacture tv ?
9.00E+00	0	who invent tv ?
2.00E+00	0	who devise television ?
2.00E+00	0	who forge tv ?
1.00E-02	1	who invent video ?
1.00E-02	1	who invent telly ?
1.00E-02	1	who contrive television ?
1.00E-02	1	who contrive tv ?
When was Babe Ruth born ?		
6.06E+03	0	when be babe ruth bear ?
3.39E+04	0	when be babe ruth pay ?
1.97E+04	0	when be babe ruth stand ?
1.09E+04	0	when be babe ruth hold ?
2.42E+03	0	when be babe ruth carry ?
1.21E+03	0	when be babe ruth have ?
4.24E+02	1	when be babe ruth support ?
9.09E+01	1	when be babe ruth expect ?
6.06E+00	1	when be babe ruth brook ?
6.06E+00	1	when be babe ruth wear ?
3.03E-01	2	when be babe ruth deliver ?
How tall is the giraffe ?		
4.00E+00	0	how tall be the giraffe ?
2.00E+00	0	how large be the giraffe ?
2.00E+00	0	how big be the giraffe ?
2.00E+00	0	how high be the giraffe ?
1.00E-01	1	how grandiloquent be the giraffe ?
1.00E-01	1	how magniloquent be the giraffe ?
1.00E-01	1	how improbable be the giraffe ?
1.00E-01	1	how marvelous be the giraffe ?

6 Evaluation

For our evaluation, we performed two retrieval tasks on the TREC LA Times collection, using TREC judgments to identify the queries that had relevant

documents in this collection. Our main evaluation was performed for the TREC-9 question-answering task, since our ultimate goal is to answer questions posed to an Internet resource. From a total of 131,896 documents in the collection, 1211 documents contained the correct answer for 404 of the 693 TREC-9 queries. An additional evaluation was performed for the TREC-6 ad-hoc retrieval task, where 1105 documents were judged relevant to 48 of the 50 TREC-6 keyword-based queries.

Our results show that query paraphrasing improves overall retrieval performance. For the ad-hoc task, when 20 retrieved documents were retained for each query, 22 correct documents in total were retrieved without paraphrasing, while a maximum of 20 paraphrases per query yielded 35 correct documents (only 18 of the 48 queries were paraphrased). For the question answering task, under the same retrieval conditions, recall improved from 294 correct documents without paraphrasing to 337 with a maximum of 20 paraphrases per query. Specifically, the number of queries for which correct documents were retrieved improved from 169 to 182.

In addition, we tested the effect of the following factors on retrieval performance.

- *WordNet co-locations* – three usages of word co-locations (none, for scoring only, for scoring and paraphrase generation).
- *Tagging accuracy* – manually-corrected tagging versus automatic PoS tagging (Brill, 1992), which tagged correctly 84% of the queries.
- *Out-of-order weight* (W_{order}) – how much we should take into account the word order in a query (strict consideration, ignore word order, intermediate).
- *Absent adjacent-pair divisor* (*AbsAdjDiv*) – how much we should penalize lemma-pairs that are adjacent in the query but absent from the corpus (same penalty as non-adjacent absent lemma-pairs, a little higher, a lot higher).
- *Query length* – how the number of words in the query affects retrieval performance.

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 1 paraphrase (Set 1), then the query plus 2 paraphrases (Set 2), and so on, up to a maximum of 19 paraphrases (Set 19). For each submission, we varied the number of documents returned by the retrieval engine from 1 to 20 documents.

6.1 WordNet Co-locations

As indicated above, we considered three usages of WordNet with respect to word co-locations: Col, NoCol and ColScore. Under the Col setting, our mechanism checked whether a lemma-pair in the input query corresponds to a WordNet co-location, and if so, generated synonyms for the pair, instead of the individual lemmas. For instance, given the lemma-pair "folic acid", the Col setting yielded synonyms such as "folate" and "vitamin m" for the lemma-pair. During paraphrase scoring, these co-locations were assigned a high frequency score, corresponding to the 999th percentile of pair frequencies in the corpus. In contrast, the NoCol setting did not take into account WordNet co-locations at all. For instance, one of the paraphrases generated by this method for "folic acid" was "folic lsd". ColScore is a hybrid setting, where WordNet was used for scoring lemma-pairs in the proposed paraphrases, but not for generating them.

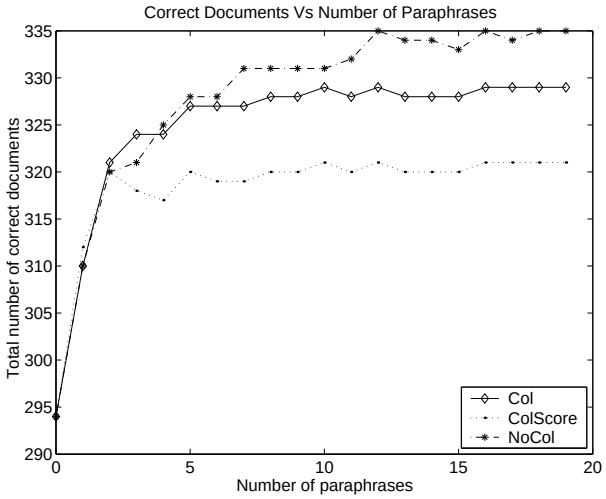

Figure 1: Effect of word co-location and number of paraphrases (20 retrieved documents)

Figure 1 depicts the total number of correct documents retrieved (for 20 retrieved documents per query), for each of the three co-location settings, as a function of the number of paraphrases in a set (from 0 to 19). The values for the other factors were: W_{order}=1, *AbsAdjDiv*=2, and manually-corrected tagging. 294 correct documents were retrieved when only the lemmatized query was submitted for retrieval (0 paraphrases). This number increases dramatically for the first few paraphrases, and eventually levels out for about 12 paraphrases. In order to compare queries that had different numbers of paraphrases, when the maximum number of paraphrases for a query was less than 19, the results obtained for this maximum number were replicated for the paraphrase sets of higher cardinality. For instance, if only 6 paraphrases were generated for a query, the number of correct documents retrieved

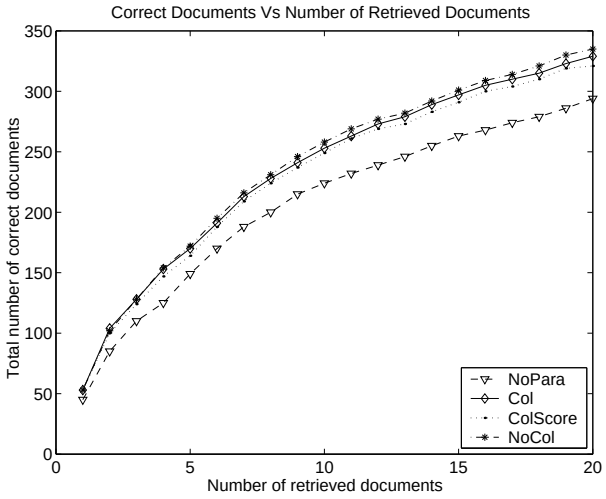

Figure 2: Effect of word co-location and number of retrieved documents (maximum paraphrases)

for the 6 paraphrases was replicated for Sets 7 to 19.

Figure 2 depicts the total number of correct documents retrieved (for 19 paraphrases or maximum paraphrases), for each of the three co-location settings, as a function of the number of documents retrieved per query (from 1 to 20). As for Figure 1, paraphrasing improves retrieval performance. In addition, as expected, recall performance improves as more documents are retrieved.

The Col setting generally yielded fewer and more felicitous paraphrases than those generated without considering co-locations (for the 118 queries where co-locations were identified). Surprisingly however, this effect did not transfer to the retrieval process, as the NoCol setting yielded a marginally better performance. This difference in performance may be attributed to whether a lemma or lemma-pair that was important for retrieval was retained in enough paraphrases. This happened in 9 instances of the NoCol setting and 2 instances of the Col setting, yielding a slightly better performance for the NoCol setting overall. For example, the identification of "folic acid" as a co-location led to synonyms such as "vitamin m" and "vitamin bc", which appeared in most of the paraphrases. As a result, the effect of the lemma-pair "folic acid", which was actually responsible for retrieving the correct document, was obscured. In contrast, the recognition of "major league" as a co-location (which was paraphrased to "big league" in only 3 of the 19 paraphrases) enabled the retrieval of the correct document. Since the performance under the ColScore condition was consistently worse than the performance under the other two conditions, we do not consider it in the rest of our evaluation.

6.2 Tagging accuracy

The PoS-tagger incorrectly tagged 64 of the 404 queries in our corpus (usually, one word was mis-tagged in each of these queries). The instances of mis-tagging which had the largest impact on the quality of the generated paraphrases occurred when nouns were mis-tagged as verbs and vice versa (18 cases). In addition, proper nouns were mis-tagged as other PoS and vice versa in 24 cases, and the verb "name" (e.g., "Name the highest mountain") was mis-tagged as a noun in 17 instances. Surprisingly, retrieval performance was affected only in 5 instances both for the Col and the NoCol settings: 3 of these instances had a mis-tagged "name", and 2 had a noun mis-tagged as another PoS.

6.3 Out-of-order weight

We considered three settings for the out-of-order weight, W_{order} (Equation 3): 1, 0 and 0.5. The first setting ignores word order. For instance, given the query "how many dogs pull a sled in the Iditarod?" the frequency of the lemma-pair "dog-pull" is added to that of the pair "pull-dog". The second setting enforces a strict word order, e.g., only "dog-pull" is considered. The third setting considers out-of-order lemma-pairs, but gives their frequency half the weight of the ordered pairs.

Interestingly, this factor had no effect on retrieval performance. This may be explained by the observation that the lemma order in the queries reflects their order in the corpus. Thus, when an ordered lemma-pair in a query matches a dictionary entry, the additional frequency count contributed by the reverse lemma order is often insufficient to affect significantly the relative score of the paraphrases.

6.4 Penalty for absent adjacent lemma-pairs

We considered four settings for the penalty assigned to lemma-pairs that are adjacent in a paraphrase but absent from the dictionary. These settings are represented by the values 1, 2, 10 and 20 for the divisor *AbsAdjDiv*. For instance, a value of 10 means that the score for an absent adjacent lemma-pair is 1/10 of the score of an absent non-adjacent lemma-pair. That is, the score of a paraphrase is divided by 100 for each absent adjacent lemma-pair.

This factor had only a marginal effect on retrieval performance, with the best performance being obtained for *AbsAdjDiv* = 10.

6.5 Query Length

Our investigation of the effect of query length on retrieval performance indicates that better performance is obtained for shorter queries. Figure 3 shows the percentage of queries where at least one correct document was retrieved, as a function of

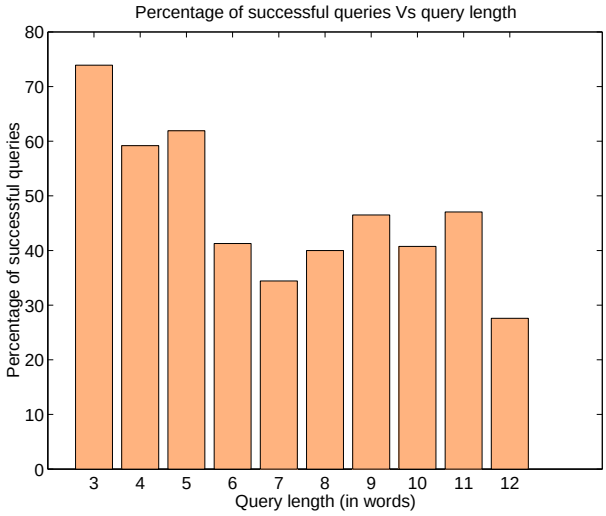

Figure 3: Effect of query length (20 retrieved documents and maximum paraphrases)

query length in words (20 documents were retrieved using 19 or maximum paraphrases). These results were obtained for the settings Col, $W_{order} = 1$ and *AbsAdjDiv*=10, with manually-corrected tagging. As seen in Figure 3, there is a drop in retrieval performance for queries with more than 5 words. These results generally concur with the observations in (Sanderson, 1994; Gonzalo et al., 1998). Nonetheless, on average we returned a correct document for 42% of the queries which had 6 to 11 words.

7 Conclusion

We have offered a mechanism for the generation of lexical paraphrases of queries posed to an Internet resource. These paraphrases were generated using WordNet and part-of-speech information to propose synonyms for the content lemmas in the queries. Statistical information obtained from a corpus was used to rank the paraphrases. Our evaluation shows that paraphrasing improves retrieval performance. This is achieved despite mis-tagging and erroneous paraphrasing of co-located words.

References

E. Brill. 1992. A simple rule-based part of speech tagger. In *ANLP-92 – Proceedings of the 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT.

C. Buckley, G. Salton, J. Allan, and A. Singhal. 1995. Automatic query expansion using SMART. In D. Harman, editor, *The Third Text REtrieval Conference (TREC3)*. NIST Special Publication.

J. Gonzalo, F. Verdejo, I. Chugur, and J. Cigarran. 1998. 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.

S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. 2001. 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.

D. Lin. 1998. 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.

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

R. Mihalcea and D. Moldovan. 1999. 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.

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

M. Mitra, A. Singhal, and C. Buckley. 1998. Improving automatic query expansion. In *SIGIR'98 – Proceedings of the 21th ACM International Conference on Research and Development in Information Retrieval*, pages 206–214, Melbourne, Australia.

G. Salton and M.J. McGill. 1983. *An Introduction to Modern Information Retrieval*. McGraw Hill.

M. Sanderson. 1994. Word sense disambiguation and information retrieval. In *SIGIR'94 – Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, pages 142–151, Dublin, Ireland.

H. Schütze and J.O. Pedersen. 1995. 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.