

Non-Contiguous Word Sequences for Information Retrieval

Antoine Doucet and Helena Ahonen-Myka

Department of Computer Science

P.O. Box 26 (Teollisuuskatu 23)

FIN-00014 University of Helsinki,

Finland,

Antoine.Doucet@cs.helsinki.fi, Helena.Ahonen-Myka@cs.helsinki.fi

Abstract

The growing amount of textual information available electronically has increased the need for high performance retrieval. The use of phrases was long seen as a natural way to improve retrieval performance over the common document models that ignore the sequential aspect of word occurrences in documents, considering them as “bags of words”. However, both statistical and syntactical phrases showed disappointing results for large document collections.

In this paper we present a recent type of multi-word expressions in the form of *Maximal Frequent Sequences* (Ahonen-Myka, 1999). Mined phrases rather than statistical or syntactical phrases, their main strengths are to form a very compact index and to account for the sequentiality and adjacency of meaningful word co-occurrences, by allowing for a gap between words.

We introduce a method for using these phrases in information retrieval and present our experiments. They show a clear improvement over the well-known technique of extracting frequent word pairs.

1 Introduction

The constantly growing number of electronic documents increases the need for high performance retrieval, the precision of a system being the percentage of relevant documents among the total number of hits returned to a query.

Most information retrieval systems do not account for word order in a document. However, we can assume that there must exist a way to account for word order, which permits to improve retrieval performance. Zhai et al. (1997) mention many problems due the use of single word terms only. They observe that some word associations have a totally different meaning of the “sum” of the meanings of the words that compose them (e.g., “hot dog” is usually not used

to refer to a warm dog !). Other lexical units pose similar problems (e.g., “kick the bucket”).

Work on the use of phrases in IR has been carried out for more than 25 years. Early results were very promising. However, unexpectedly, the constant growth of test collections caused a drastic fall in the quality of the results. In 1975, Salton et al. (1975) show an improvement in average precision over 10 recall points between 17% and 39%. In 1989, Fagan (1989) reiterated the exact same experiments with a 10 Mb collection and obtained improvements from 11% to 20%. This negative impact of the collection size was lately confirmed by Mitra et al. (1987) over a 655 Mb collection, improving the average precision by only one percent ! Turpin and Moffat (1999) revisited and extended this work to obtain improvements between 4% and 6%.

A conclusion of this related work is that phrases improve results in low levels of recall, but are globally inefficient for the n first ranked documents. According to Mitra et al. (1987), this low benefit from phrases to the best answers is explained by the fact that phrases promote documents that deal with only one aspect of possibly multi-faceted queries. For example, a topic of TREC-4 is about “problems associated with pension plans, such as fraud, skimming, tapping or raiding”. Several top-ranked documents discuss pension plans, but no related problem. Mitra et al. (1987) term this problem as one of *inadequate query coverage*.

In our opinion, this does not contradict the idea that adding document descriptors accounting for word order must permit to improve the performance of IR systems. But related work shows the need for another way to combine phrase and word term descriptors (Smeaton and Kelledy, 1998) and even more the fact that the phrases currently used to model documents are not well suited for that.

In the next section, we will briefly describe

the vector space model (sometimes quoted as “bag of words”, for it simply ignores words’ positions). We will then describe the different types of phrases used in related work (section 3). In section 4, we define our own phrases (maximal frequent sequences) and explain how they will be better document descriptors than those found in the state of the art. In section 5, we present a technique to incorporate maximal frequent sequences into document indexing and query processing, so as to properly take advantage of this extra information in an information retrieval framework. In section 6, we present our experiments and results, before we conclude the paper in section 7.

2 Vector Space Model

2.1 Preprocessing

The first step of the process is to clean the data. A way to do this consists in skipping a set of words that are considered least informative, the *stopwords*. We also discarded all words of small size (less than three characters).

We then reduced each word to its stem using the Porter algorithm (Porter, 1980). For example, the words “models”, “modelling” and “modeled” are all stemmed to “model”. This technique for reducing words to their root permits to further reduce the number of word terms.

This feature selection phase brings more computational comfort for the next steps since it greatly reduces the size of the document collection representation in the vector space model (the *dimension* of the vector space).

2.2 Vector Space Model

The set of the distinct remaining word stems W is used to represent the document collection within the *vector space model*. Each document is represented by a $\|W\|$ -dimensional vector filled in with a weight standing for the importance of each word token with respect to that document. To calculate this weight, we use a *tf*-normalized version of the “*tfc*” term-weighted components as described by Salton and Buckley (1988), i.e.:

$$tfidf_w = \frac{tf_w \cdot \log \frac{N}{n_w}}{\max(tf) \cdot \sqrt{\sum_{w_i \in W} \left(tf_{w_i} \cdot \log \frac{N}{n_{w_i}} \right)^2}},$$

where tf_w is the term frequency of the word w . N is the total number of documents in the

collection and n_w the number of documents in which w occurs.

3 The use of phrases in IR

There are various ways to incorporate phrases in the document modeling. The usual technique is to consider phrases as supplementary terms of the vector space, with the same technique as for word terms. In other words, phrases are thrown into the bag of words. However, Strzalkowski and Carballo (1996) argue that using a standard weighting scheme is inappropriate for mixed feature sets (such as single words and phrases). The weight given to least frequent phrases is considered too low. Their specificity is nevertheless often crucial in order to determine the relevance of a document (Lahtinen, 2000). In weighting the phrases, the interdependency between a phrase and the words that compose it is another difficult issue to account for Strzalkowski et al. (1998).

There are two main types of phrases: statistical phrases, formed by straight word occurrence counts, and syntactical phrases.

Statistical Phrases. Mitra et al. (1987) form a statistical phrase for each pair of 2 stemmed adjacent words that occur in at least 25 documents of the TREC-1 collection. The selected pairs are then sorted in lexicographical order. In this technique, we see 2 problems. First, this lexicographical sorting means to ignore crucial information about word pairs: their order of occurrence ! This is equivalent to saying that AB is identical to BA. Furthermore, no gap is allowed, although it is frequent to represent the same concept by adding at least one word between two others. For example, this definition of a phrase does not permit to note any similarity between the two text fragments “XML document retrieval” and “XML retrieval”. This model is thus quite far from natural language.

Syntactical Phrases. The technique presented by Mitra et al. (1987) for extracting syntactical phrases is based on a parts-of-speech analysis (POS) of the document collection. A set of tag sequence patterns are predefined to be recognized as useful phrases. All maximal sequences of words accepted by this grammar form the set of syntactical phrases. For example, a sequence of words tagged as “verb, cardinal number, adjective, adjective, noun” will constitute a syntactical phrase of size 5. Every sub-phrase occurring in this same order is also

generated, with an unlimited gap (e.g., the pair “verb, noun” is also generated). This technique offers a sensible representation of natural language. Unfortunately, to obtain the POS of a whole document collection is very costly. The index size is another issue, given that all phrases are stored, regardless of their frequency. In the experiments, the authors indeed admit to creating no index *a priori*, but instead that the phrases were generated according to each query. This makes the process tractable, but implies very slow answers from the retrieval system, and quite a long wait for the end user.

On top of computational problems, we see a few further issues. First, the lack of a minimal frequency threshold to reduce the number of phrases in the index. This means that infrequent phrases are taking up most of the space, and have a big influence on the results, whereas their low frequency may simply illustrate an inadequate use or a typographical error. To allow an illimited gap so as to generate subpairs is dangerous as well: the phrase “I like to eat hot dogs” will generate the subpair “hot dogs”, but it will also generate the subpair “like dogs”, whose semantical meaning is very far from that of the original sentence.

Other types of phrases. Many efficient techniques exist to extract multiword expressions, collocations, lexical units and idioms (Church and Hanks, 1989; Smadja, 1993; Dias et al., 2000; Dias, 2003). Unfortunately, very few have been applied to information retrieval with a deep evaluation of the results.

Maximal Frequent Sequences. We propose Maximal Frequent Sequences (MFS) as a new alternative to account for word ordering in the modeling of textual documents. One of their strength is the fact that they are extracted if and only if they occur more often than a given frequency threshold, which hopefully permits to avoid storing the numerous least significant phrases. A gap between words is allowed within the extraction process itself, permitting to deal with a larger variety of language.

4 Maximal Frequent Sequences

In our approach, we represent documents by word features within the vector space model, and by Maximal Frequent Sequences, accounting for the sequential aspect of text. For each of those two representations, a Retrieval Status Value (RSV) is computed. Those values are

later combined to form a single RSV per document.

4.1 Definition and Extraction Technique

MFS are sequences of words that are frequent in the document collection and, moreover, that are not contained in any other longer frequent sequence. Given a frequency threshold σ , a sequence is considered to be frequent if it appears in at least σ documents.

Ahonen-Myka (1999) presents an algorithm combining bottom-up and greedy methods, which permits to extract maximal sequences without considering all their frequent subsequences. This is a necessity, since maximal frequent sequences in documents may be rather long.

Nevertheless, when we tried to extract the maximal frequent sequences from the collection of documents, their number and the total number of word features in the collection did pose a clear computational problem and did not actually permit to obtain any result.

To bypass this complexity problem, we decomposed the collection of documents into several disjoint subcollections, small enough so that we could efficiently extract the set of maximal frequent sequences of each subcollection. Joining all the sets of MFS’, we obtained an *approximate* of the maximal frequent sequence set for the full collection.

We conjecture that more consistent subcollections permit to obtain a better approximation. This is due to the fact that maximal frequent sequences are formed from similar text fragments. Accordingly, we formed the subcollection by clustering similar documents together using the well-known k-means algorithm (see for example Willett (1988) or Doucet and Ahonen-Myka (2002)).

4.2 Main Strengths of the Maximal Frequent Sequences

The method efficiently extracts all the maximal frequent word sequences from the collection. From the definitions above, a sequence is said to be maximal if and only if no other frequent sequence contains that sequence.

Furthermore, a *gap* between words is allowed: in a sentence, the words do not have to appear continuously. A parameter g tells how many other words two words in a sequence can have between them. The parameter g usually gets

values between 1 and 3.

For instance, if $g = 2$, a phrase “President Bush” will be found in both of the following text fragments:

..President of the United States Bush..
 ..President George W. Bush..

Note: Articles, prepositions and small words were pruned away during the preprocessing.

This allowance of gaps between words of a sequence is probably the strongest specificity of the method, compared to most existing methods for extracting text descriptors. This greatly increases the quality of the phrase, since processing takes the variety of natural language into account.

The other powerful specificity of the technique is the ability to extract maximal frequent sequences of any length. This permits to obtain a very compact description of documents. For example, by restricting the length of phrases to 8, the presence, in the document collection, of a frequent phrase of 25 words would result in thousands of phrases representing the same knowledge as this one maximal sequence.

The result of this extraction is that each document of the collection is described by a (possibly empty) set of MFS.

5 Evaluating Documents

Once documents and queries are represented within our two models, a way to estimate the relevance of a document with respect to a query remains to be found. As mentioned earlier, we compute two separate RSV values for the word features vector space model and the MFS model. In the second step, we aggregate these two RSVs into one single relevance score for each document with respect to a query.

5.1 Word features RSV

The vector space model offers a very convenient framework for computing similarities between documents and queries. Indeed, there exist a number of techniques to compare two vectors, Euclidean distance, Jaccard and cosine similarity being the most frequently used in IR. We have used cosine similarity because of its computational efficiency. By normalizing the vectors, which we did in the indexing phase, $\text{cosine}(\vec{d}_1, \vec{d}_2)$ indeed simplifies to the vector product $(d_1 \cdot d_2)$.

5.2 MFS RSV

The first step is to create an MFS index for the document collection. Once a set of maximal frequent sequences has been extracted and each document is attached to the corresponding phrases, as detailed in the previous section, it remains to define the procedure to match a phrase describing a document and a keyphrase (from a query).

Note that from here onwards, *keyphrase* denotes a phrase found in a query, and *maximal sequence* denotes a phrase extracted from a document.

Our approach consists in decomposing keyphrases of the query into pairs. Each of these pairs is bound to a score representing its *quantity of relevance*. Informally speaking, the quantity of relevance of a word pair tells how much it makes a document relevant to include an occurrence of this pair. This value depends on the specificity of the pair (expressed in terms of inverted document frequency) and modifiers, among which is an *adjacency coefficient*, reducing the quantity of relevance given to a pair formed by two words that are not adjacent.

5.2.1 Definitions:

Let D be a collection of N documents and $A_1 \dots A_m$ a keyphrase of size m . Let A_i and A_j be 2 words of $A_1 \dots A_m$ occurring in this order, and n be the number of documents of the collection in which $A_i A_j$ was found. We define the quantity of relevance of the pair $A_i A_j$ to be:

$$Q_{rel}(A_i A_j) = idf(A_i A_j, D) \cdot adj(A_i A_j),$$

where $idf(A_i A_j, D)$ represents the specificity of $A_i A_j$ in collection D :

$$idf(A_i A_j, D) = \log \left(\frac{N}{n} \right),$$

and when decomposing the keyphrase $A_1 \dots A_m$ into pairs, $adj(A_i A_j)$ is a score modifier to penalize word pairs $A_i A_j$ formed from non-adjacent words, and $d(A_i, A_j)$ indicates the number of words appearing between the two words A_i and A_j ($d(A_i, A_j) = 0$ signifies that A_i and A_j are adjacent):

$$adj(A_i A_j) = \begin{cases} 1, & & \text{if } d(A_i, A_j) = 0 \\ \alpha_1, & 0 \leq \alpha_1 \leq 1, & \text{if } d(A_i, A_j) = 1 \\ \alpha_2, & 0 \leq \alpha_2 \leq \alpha_1 & \text{if } d(A_i, A_j) = 2 \\ \dots & & \\ \alpha_{m-2}, & 0 \leq \alpha_{m-2} \leq \alpha_{m-3}, & \text{if } d(A_i, A_j) = m-2 \end{cases}$$

Accordingly, the larger the distance between the two words, the lower a quantity of relevance is attributed to the corresponding pair. In our runs, we will actually ignore distances higher than 1 (i.e., $(k > 1) \Rightarrow (\alpha_k = 0)$).

5.2.2 Example:

For example, ignoring distances above 1, a keyphrase ABCD is decomposed into 5 tuples (pair, adjacency coefficient):

$$(AB, 1), (BC, 1), (CD, 1), (AC, \alpha_1), (BD, \alpha_1)$$

Let us compare this keyphrase to the documents d_1, d_2, d_3, d_4 and d_5 , described respectively by the frequent sequences AB, AC, AFB, ABC and ACB. The corresponding quantities of relevance brought by the keyphrase ABCD are shown in table 1. Note that in practice, we lost the maximality property during the partition-joint step presented in subsection 4.1. Hence, there can be a frequent sequence AB together with a frequent sequence ABC, if they were extracted from two different document clusters.

Assuming equal *idf* values, we observe that the quantities of relevance form a coherent order. The longest matches rank first, and matches of equal size are untied by adjacency. Moreover, non-adjacent matches (AC and ABC) are not ignored as in many other phrase representations (Mitra et al., 1987).

5.3 Aggregated RSV

In practice, some queries do not contain any keyphrase, and some documents do not contain any MFS. However, there can of course be correct answers to these queries, and those documents must be relevant to some queries. Also, all documents containing the same matching phrases get the same MFS RSV. Therefore, it is necessary to find a way to separate them. The word-based cosine similarity measure is very appropriate for that.

Another natural response would have been to re-decompose the pairs into single words and form document vectors accordingly. However, this would not be satisfying, because the least frequent words are all missed by the algorithm for MFS extraction. An even more important category of missed words is that of frequent words that do not frequently co-occur with other words. The loss would be considerable.

This is the reason to compute another RSV using a basic word-features vector space model.

```
<Keywords>
"concurrency control"
"semantic transaction management"
"application" "performance benefit"
"prototype" "simulation" "analysis"
</Keywords>
```

Figure 1: Topic 47

To combine both RSVs to one single score, we must first make them comparable by mapping them to a common interval. To do so, we used *Max Norm*, as presented by Vogt and Cottrell (1998), which permits to bring all positive scores within the range [0,1]:

$$New\ Score = \frac{Old\ Score}{Max\ Score}$$

Following this normalization step, we aggregate both RSVs using a linear interpolation factor λ representing the relative weight of scores obtained with each technique (similarly as in Marx et al. (2002)).

$$Aggregated\ Score = \lambda \cdot RSV_{Word\ Features} + (1 - \lambda) \cdot RSV_{MFS}$$

The evidence of experiments with the INEX 2002 collection showed good results when weighting the single word RSV with the number of distinct word terms in the query (let a be that number), and the MFS RSV with the number of distinct word terms found in keyphrases of the query (let b be that number). Thus:

$$\lambda = \frac{a}{a + b}$$

For example, in Figure 1 showing topic 47, there are 11 distinct word terms and 7 distinct word terms occurring in keyphrases. Thus, for this topic, we have $\lambda = \frac{11}{11+7}$.

6 Experiments and Results

We based our experiments on the 494Mb INEX document collection (Initiative for the Evaluation of XML retrieval¹). INEX was created in 2002 to compensate the lack of an evaluation forum for the XML information retrieval. This collection consists of 12,107 scientific articles written in English from IEEE journals, combined to a set of queries and corresponding manual assessments. The specificity of this

¹available at <http://inex.is.informatik.uni-duisburg.de:2003/>

Document	MFS	Corresponding pairs	Matches	Quantity of relevance
d_1	AB	AB	AB	idf(AB)
d_2	ACD	AC CD AD	AC CD	idf(CD) + α_1 .idf(AC)
d_3	AFB	AF FB AB	AB	idf(AB)
d_4	ABC	AB BC AC	AB BC AC	idf(AB) + idf(BC) + α_1 .idf(AC)
d_5	ACB	AC CB AB	AC AB	idf(AB) + α_1 .idf(AC)

Table 1: Quantity of relevance stemming from various indexing phrases w.r.t. a keyphrase query ABCD

document collection is its rich logical structure into sections, subsections, paragraphs, lists, etc. However, in the present experiments, we ignore this structure and only exploit plain text to return full articles as our candidate retrieval answers.

The manual assessments indeed tell us which candidate answers are relevant and which ones are not. We use these relevance values to compute precision and recall measures, which permit scoring each set of candidate answers, and equivalently the means by which each set was obtained. In our experiments, we used average precision over the n first hits as our main reference. This evaluation measure was first introduced by Raghavan et al. (1989) and was used as the official evaluation measure in the INEX 2002 campaign (Gövert et al., 2003).

Protocol of the Experiments. As a baseline, we computed and evaluated a run using only single word terms, as detailed in section 2. Our goal was to compare our new technique to the state of the art. Thus we computed one run using our technique (aggregating the MFS RSVs and the single word term RSVs topic-wise, with the weighting scheme mentioned hereabove), and one run by calculating all statistical phrases following the definition of Mitra et al. (1987). The only difference is that we did not set a minimal document frequency threshold. We made this choice from the standpoint that our aim was not to measure efficiency, but the quality of the results. The corresponding number of features is given in table 2. We extracted 328,289 MFS of different sizes. Their splitting forms no more than 674,257 pairs (this number is probably lower because the same pair can be extracted from numerous MFS).

MFS vs. Statistical Phrases. For those representations, the average precision for the n first retrieved documents are presented in table 3. We learn two things from those results.

	Number of Features
Word terms (Baseline)	156,723
Statistical Phrases	4,941,051
MFS	674,257

Table 2: Number of feature terms

Weight of the word RSV	Words & Stat. Pairs
Topicwise (subsection 5.3.)	0.05825
20%	0.05902
40%	0.05957
60%	0.05843
80%	0.05527
100%	0.05302

Table 4: Average Precision@100 for various linear combinations

First, the fact that phrases improve results in lower levels of recall is confirmed, as greater improvement is obtained when we check further down the ranked list. Second, our technique outperforms that of statistical phrases. However, as we use different phrases indeed, but also a different technique to match them against queries, it remains to find out whether the improvement stems from the MFS themselves, from the way they are used, or from both.

Thus we experimented with various linear combinations to aggregate the word term RSV and the statistical phrase RSV. The results are presented in table 4. The technique of gathering word and pairs features within the same vector space clearly performs better in this case. Therefore, the better performance of MFS is not only due to the aggregation weighting scheme presented in subsection 5.3. This underlines their intrinsic quality as document descriptors.

	Word Terms	Words and Stat. Phrases	Words and MFS
Average Precision@100	0.05302	0.06199 (+16.9%)	0.06713 (+26.6%)
Average Precision@50	0.64419	0.62456 (-3.0%)	0.64411 (-0.0%)
Average Precision@10	0.67101	0.65021 (-3.1%)	0.66293 (-1.2%)

Table 3: Average Precision@n

7 Conclusions

We have introduced a new type of phrases to the problem of information retrieval. We have developed and presented a method to use maximal frequent sequences in information retrieval. Using the INEX document collection, we compared it to a well-known technique of the state of the art. Our technique outperformed that of statistical phrases, known to be performing comparably to syntactical and linguistic phrases from the literature.

These results are due to the allowance of a gap between words forming a sequence, offering a more realistic model of natural language. Furthermore, the number of phrases to index is rather small. A weak spot is the greedy algorithm to extract MFS. But many improvements are under way on this side, and the partition-join technique mentioned in subsection 4.1 already permits to extract good approximations efficiently.

Our results confirm that the best improvements are obtained at the highest levels of recall. Therefore, MFS would be most useful in the case of *exhaustive* information needs. Cases where no relevant information should be missed, and 100% recall should be reached in a minimal number of hits (their inner ordering being a less serious matter). Typically, examples of such information lie in the judicial domain and in patent searching.

More experiments remain to be done, to find out whether similar improvements can be obtained from other document collections. The INEX collection is of scientific articles and consistently uses a terminology of its own. Whether similar performance would be observed from a more general document collection such as newspaper articles has to be verified.

The use of phrases is factual in many languages, which makes us optimistic regarding an application of this work to multilingual document corporas. Thinking of the other techniques, the gap should give us robustness against the challenges of multilingualism.

8 Acknowledgements

This work was funded by the Academy of Finland under project 50959: DoReMi - Document Management, Information Retrieval and Text Mining.

References

- Helena Ahonen-Myka. 1999. Finding All Frequent Maximal Sequences in Text. In *Proceedings of the 16th International Conference on Machine Learning ICML-99 Workshop on Machine Learning in Text Data Analysis, Ljubljana, Slovenia*, pages 11–17. J. Stefan Institute, eds. D. Mladenic and M. Grobelnik.
- Kenneth W. Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th meeting of the Association for Computational Linguistics (ACL)*, pages 76–83.
- Gaël Dias, Sylvie Guilloché, Jean-Claude Basano, and José Gabriel Pereira Lopes. 2000. Combining linguistics with statistics for multiword term extraction: A fruitful association? In *Proceedings of Recherche d’Informations Assistée par Ordinateur 2000 (RIAO 2000)*.
- G. Dias. 2003. Multiword unit hybrid extraction. In *Workshop on Multiword Expressions of the 41st ACL meeting, Sapporo, Japan*.
- A. Doucet and H. Ahonen-Myka. 2002. Naive clustering of a large xml document collection. In *Proceedings of the First Workshop of the Initiative for the Evaluation of XML Retrieval (INEX)*, pages 81–87, Schloss Dagstuhl, Germany.
- J. L. Fagan. 1989. The effectiveness of a non-syntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science*, 40:115–132.
- Norbert Gövert, Gabriella Kazai, Norbert Fuhr, and Mounia Lalmas. 2003. Evaluating the effectiveness of content-oriented XML retrieval. Technical report, University of Dortmund, Computer Science 6.

- Timo Lahtinen. 2000. *Automatic Indexing: an approach using an index term corpus and combining linguistic and statistical methods*. Ph.D. thesis, University of Helsinki.
- M. Marx, J. Kamps, and M. de Rijke. 2002. The university of amsterdam at inex.
- M. Mitra, C. Buckley, A. Singhal, and C. Cardie. 1987. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO97, Computer-Assisted Information Searching on the Internet*, pages 200–214.
- M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.
- V. V. Raghavan, P. Bollmann, and Jung G. S. 1989. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems*, 7(3):205–229.
- G. Salton and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24(5):513–523.
- G. Salton, C.S. Yang, and C.T. Yu. 1975. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 26:33–44.
- F. Smadja. 1993. Retrieving collocations from text: Xtract. *Journal of Computational Linguistics*, 19:143–177.
- A. F. Smeaton and F. Kelledy. 1998. User-chosen phrases in interactive query formulation for information retrieval. In *Proceedings of the 20th BCS-IRSG Colloquium*.
- Tomek Strzalkowski and Jose Perez Carballo. 1996. Natural language information retrieval: TREC-4 report. In *Text REtrieval Conference*, pages 245–258.
- Tomek Strzalkowski, Gees C. Stein, G. Bowden Wise, Jose Perez Carballo, Pasi Tapanainen, Timo Jarvinen, Atro Voutilainen, and Jussi Karlgren. 1998. Natural language information retrieval: TREC-6 report. In *Text REtrieval Conference*, pages 164–173.
- A. Turpin and A. Moffat. 1999. Statistical phrases for vector-space information retrieval. In *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 309–310.
- Christopher C. Vogt and Garrison W. Cottrell. 1998. Predicting the performance of linearly combined IR systems. In *Research and Development in Information Retrieval*, pages 190–196.
- P. Willett. 1988. Recent trends in hierarchical document clustering: a critical review. *Information Processing and Management*, 24(5):577–597.
- Zhai, Chengxiang, Xiang Tong, N. Milic Frayling, and Evans D.A. 1997. Evaluation of syntactic phrase indexing. In *Proceedings of the 5th Text Retrieval Conference, TREC-5*, pages 347–358.