SICS at CLEF 2002: Automatic Query Expansion Using Random Indexing

Magnus Sahlgren⋆, Jussi Karlgren⋆, Rickard Cöster⋆ and Timo Järvinen°

⋆Swedish Institute of Computer Science, SICS,
Box 1263, SE-164 29 Kista, Sweden
{*mange, jussi, rick*}*@sics.se*

°Conexor oy,
Helsinki Science Park, Koetilantie 3, 00710 Helsinki, Finland
timo@conexor.fi

Abstract

Vector-space techniques can be used for extracting semantically similar words from the co-occurrence statistics of words in large text data. We have used a technique called Random Indexing to accumulate context vectors for Swedish, French and Italian. We have then used the context vectors to perform automatic query expansion. In this paper, we report on our CLEF 2002 experiments on Swedish, French and Italian monolingual query expansion.

1 Introduction: Queries and Query Expansion

Arguably, query formulation is the major bottleneck for satisfying user needs in information access applications. This problem is aggravated in a cross-lingual context; finding the right word in a non-native language is even more of a problem than in one's first language.

The fundamental problem in the query formulation process is what has become known as the *vocabulary* problem, since it concerns people's choice of words to express their information need. There are two facets to this dilemma; the *synonymy* and *polysemy* problems. The former is the problem that people might choose different words to express the same information. For example, one person might use the word "boat" to refer to water crafts, while another person might use the word "ship". The polysemy problem, on the other hand, is the problem that one "word" (or, rather, one orthographic construction) can have several meanings — the cardinal example here being "bank" (as in "sandbank", "monetary institution" etc.). In other words, words are both too specific and too vague at the same time. If the retrieval system does not attempt to handle these vocabulary discrepancies, it might miss relevant documents, or, even worse, it might retrieve totally irrelevant documents.

One common solution to the vocabulary problem is to add additional terms to the query. This methodology is generally known as *query expansion*, and it can be used to tackle both the synonymy and polysemy problems. For example, we can add additional terms to a query in order to disambiguate a polysemous word (e.g. if the query contains "bank", it might be wise to add an additional term specifying whether it is sandbanks (e.g. "sand") or monetary institutions (e.g. "money") that is being referred to), or, which is the more common reason for performing a query expansion, we can add additional terms to a query in order to ensure adequate recall for the query (e.g. by adding "ship" to a query containing "boat").

Our approach to handle the vocabulary problem is to use a statistical technique for automatic query expansion. The idea is to use a vector-space model to extract semantically similar words from the co-occurrence statistics of words in large text data. Thus, if “boat” is in the query, the system should expand the query with “ship,” “vessel,” “craft,” “water” and so on. In this paper, we report on our CLEF 2002 experiments with using statistically based automatic query expansion.

1.1 An Overview of the Vector-Space Methodology

Vector-space models use co-occurrence statistics to generate word vectors that can be used to calculate similarity between words. Traditionally, this is done by representing the text data as an $n \times m$ co-occurrence matrix, where each row n represents a unique word and each column m represents a document or a word. For example, Latent Semantic Analysis/Latent Semantic Indexing (LSA/LSI) [8], [2] uses a words-by-document matrix, whereas Hyperspace Analogue to Language (HAL) [9] uses a words-by-words matrix. The cells of the co-occurrence matrix are the (normalized) frequency counts of a particular word in a particular context (document or word). The rows of the co-occurrence matrix can be interpreted as *context vectors* for the words in the vocabulary, making it straight-forward to express the distributional similarity between words in terms of vector similarity.

We have chosen a somewhat different methodology to construct the co-occurrence matrix. The technique, which we call *Random Indexing* [5], [6], uses *distributed* representations to accumulate context vectors from the distributional statistics of words. This is accomplished by first assigning a unique high-dimensional sparse random *index vector* to each word type in the text data. Then, every time a word occurs in the text data, we add the index vectors of the n surrounding words to the context vector for the word in question. Mathematically, this technique is equivalent to the Random Mapping approach described in [7].

The resulting high-dimensional context vectors thus represent the distributional profiles of words, by effectively being the sum of (the representations of) every word that the target word has co-occurred with. Now, according to the *Distributional Hypothesis*, which states that two words are semantically similar to the extent that they share contexts [4], the context vectors can be used to calculate (distributional) similarity between words. We do this by simply calculating the cosine of the angles between the context vectors.

2 The CLEF 2002 Query Expansion Experiments

In the CLEF 2002 monolingual retrieval task, we have used Random Indexing to construct context vectors for words in Swedish, French and Italian. We have then used the context vectors to extract the nearest neighbors (i.e. the most correlated terms) to words in the CLEF queries. In effect, what we have produced is a series of automatically generated thesauri.

2.1 Training Data

The system was trained using the CLEF 2002 collections for Swedish, French and Italian. The data files were preprocessed and morphologically analyzed and normalized to base form using syntactic analysis tools from Conexor.[1] The same morphological analysis and normalization procedure was applied to the topic texts.

2.2 Applying the Random Indexing Technique

In order to generate context vectors for the words in the Swedish, French and Italian training data, we first assigned a 1,800-dimensional sparse random index vector to each word type in the three different training data. The 1,800-dimensional random index vectors consisted of 8 randomly

[1] http://www.conexor.fi

distributed -1s and $+1$s (4 of each), with the rest of the elements in the vectors set to zero. We used these parameters since they have been verified in other experiments to be viable for extracting semantically related words from co-occurrence information [6]. However, it should be noted that the technique does perform better the higher the dimensionality of the vectors are. Of course, there is a trade-off between performance and efficiency, since using very high-dimensional vectors would be computationally demanding.

As previously described, the 1,800-dimensional random index vectors were then used to accumulate context vectors for the words. This was done by adding the index vectors of the n surrounding words to the context vector for a given word every time the word occurred in the training data. In these experiments, we used $n = 6$ (i.e. the three preceding and the three succeeding words), since this has proven to be a viable context size to capture distributional distinctiveness [6]. Also, the context windows were weighted by the function 2^{1-d}, where d is the distance (in word units) to the target word. Finally, we used a frequency threshold for the vector additions to exclude words with a frequency less than 3, since low frequency words give unreliable statistical estimates.

2.3 Query Construction and Expansion

The queries were constructed by removing stop words and some query specific terms (e.g. "find", "documents", "relevant" etc.) from the <title> and <description> fields of the CLEF 2002 topics.

For the query expansion runs, we expanded every word in the queries by the 5 words whose context vectors were most similar to the context vector of the original query word. Vector similarity was computed as the cosine of the angle between the context vectors:

$$d_{cos}(x,y) = \frac{\vec{x} \cdot \vec{y}}{\mid \vec{x} \mid\mid \vec{y} \mid} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

This measure gives an estimate of the amount of correlation between the vectors, ranging from 1 (which is a perfect match) to -1 (which means that the vectors are totally uncorrelated).[2] To ensure that only words with a high degree of correlation to the original query word were included in the expanded queries, we used a threshold for the correlations at 0.2.

3 Search engine

The text retrieval engine used for our experiments is the first version of a system being developed at SICS. It currently supports Vector Space, Boolean and structured queries. In the following sections, we provide a detailed description of the index structure and query methods.

3.1 Document format

Documents are converted from their original format to a common XML format prior to indexing. Each XML document contains several meta-data tags such as creation date, author and original location as well as a list of sections (structured fields). A section has a name, a weight and a block of textual content. The XML file as well as the index is stored in Unicode format, a prerequisite for cross-language indexing and retrieval.

We use a simple procedure for term extraction from the XML documents, since the document content has already been formatted by lexical analysis and stemming.

3.2 Index construction

For the inverted index, we use a Simple Prefix B^{+}-tree [1], [3]. To construct the index and the inverted lists, we use sort-based inversion [12] where the sort phase is implemented as an external

[2] In practice, two distributionally unrelated words get a correlation round 0. Negative correlations are merely the result of noise.

(disk-based) k-way merge sort [3], [12]. The inverted lists of document id and term frequency pairs $(d_i, tf_{d_i,t})$ are compressed for each term t. The d_i numbers are run-length encoded and the $(d_i, tf_{d_i,t})$ pairs are further compressed using integer compression.

A number of different integer compression algorithms have been proposed in the literature: Golomb coding, Elias delta coding, Elias gamma coding etc. [11], [12]. We use Golomb coding for the d_i values and Elias delta coding for the $tf_{d_i,t}$ values.

3.3 Queries

The query language supports Vector Space, Boolean and structured queries. All queries in the experiments were evaluated as full-text Vector Space queries.

3.3.1 Vector Space queries

For Vector Space queries, we use an approximate cosine measure, where documents are normalized by the square root of the number of terms in the document, instead of the Euclidean length of the document's vector of $tf * idf$ values. The similarity between a query q and document d is

$$sim(q,d) \quad = \quad \frac{\sum_{t \in T_q} w_{t,q} * w_{t,d}}{\sqrt{\text{number of terms in } d}}$$

where T_q is the set of terms in the query.

The term weighting scheme $w_{t,d}$ for term t in document d follows a classical model of the product of term frequency tf and inverse document frequency idf:

$$tf * idf \quad = \quad (0.5 + 0.5 \frac{tf_{t,d}}{\max tf_d}) * \log_2 \frac{N}{n(t)}$$

where N is the total number of documents in the collection and $n(t)$ is the number of documents containing term t. The query term weight $w_{t,q}$ is set to 1 in our experiments.

4 Results

We used the preprocessed Swedish, French and Italian CLEF 2002 collections for the automatic monolingual task. A separate index was created for each language. Queries were submitted twice to the retrieval engine; with and without query expansion.

The top 1000 documents were taken as the result list for each query. We report the non-interpolated average precision as well as R-Precision scores for each run in Table 4. The suffix X for languages denote the automatic semantic query expansion result for the query. Rel. is the number of relevant documents for the queries, Ret. is how many relevant documents we found. Avg. p. is the non-interpolated average precision for all relevant documents, R-Prec. is the precision after our system had retrieved Rel. number of documents.

Table 1: Results from SICS automatic monolingual runs

	Rel	**Ret**	**Avg. p.**	**R-Prec.**
Swedish	1196	836	0.1347	0.1432
SwedishX	1196	850	0.1053	0.1170
Italian	1072	871	0.2239	0.2344
ItalianX	1072	844	0.1836	0.1799
French	1383	1117	0.2118	0.2155
FrenchX	1383	1072	0.1775	0.2030

4.1 Analysis

The results are decidedly mixed. Our expanded runs show consistently lower results than the unexpanded ones; all runs are below median on most queries, but all also score at least some query over median.

Unfortunately, the preprocessing of the Italian and French queries failed to remove most query specific terms. However, even though there were a considerable amount of noise in some of the queries (introduced by query terms and expansions of query terms), the performance of the system does not seem to be gravely affected by this. That is, there does not seem to be any consistent correlation between low retrieval performance and the presence of topically irrelevant words (i.e. noise) in the queries. This indicates that the system is very robust.

4.1.1 Lexical factors

It is clear that we need to rethink the utility of statistically based and lexically agnostic expansion. We should be able to typologize terminology so that expansion will be performed by term type rather than blindly. Person names, for instance, most likely should not be expanded to other names. Place names, on the other hand, could be expanded to hypernyms or related places names with less risk of introducing noise. To demonstrate the problem, consider query nr. 123, which features a number of person names, which get expanded with other names:

Marie → *Claude Pierre Gabin Harlow Francois*

in the Italian expansion, and

Marie → *Helsén Helsen Hedborg Cardesjö Fritthioff*

in the Swedish expansion.

Arguably, expansion of person names merely introduce noise into the query. On the other hand, consider query nr. 118, which features a place name:

Finland → *Funlandia Norvegia Islanda Svezia Danimarca*

in the Italian expansion, and

Finland → *Norge Danmark Sverige Österrike Island*

in the Swedish expansion.

These expansion terms show better promise of usefulness, as they all refer to (geographically, culturally, politically and economically) similar countries.

We will investigate the possibility of automatically understanding the lexical category of a term from its statistical properties; we would prefer to use as little hand-compiled lexical resources as possible so as not to limit the generality of the results to high-density languages.

4.1.2 Query term selection

Another important issue that we need to consider when performing automatic query expansion is the selection of query terms. In the present experiments, we simply expanded the queries word by word. As demonstrated above, this proved to be an inefficient approach, as many of the distributionally related words were clearly unsuitable as expansion terms in the given search context. This problem derives from the fact that the similarity relations between the context vectors merely reflect the distribution of words in the training data. In other words, the context vectors are highly domain specific, and reflect the topicality of the collection they were extracted from.

To demonstrate the domain specificity of the context vectors, we trained the system on two different text data: the 60-million-word Los Angeles Times TREC collection and the 10-million-word Touchstone Applied Science Associates (TASA) corpus. We then extracted the 5 nearest

Table 2: Nearest neighbors (NN) to “invasion” in 5 different corpora.

Corpus	**1st NN**	**2nd NN**	**3rd NN**	**4th NN**	**5th NN**
LA Times	withdrawal	aggression	invading	invaded	troops
TASA	revolt	discriminated	rebelled	rebelling	immunized
CLEF 2002 Swedish collection	invadera	Haiti	haitiinvasion	kuwaitkonflikt	haitijunta
CLEF 2002 Italian collection	Iraq	city	Baghdad	invadere	Haiti
CLEF 2002 French collection	Bahrain	Irak	indépence	Bahraïn	monuik

neighbors to “invasion”, and compared the results to those produced by training on the CLEF 2002 collections:

These examples clearly demonstrate the fact that the correlations extracted from co-occurrence information reflect the topicality of the training data. The LA Times and TASA corpora produce fairly generic expansion terms, whereas the CLEF 2002 collections generate highly domain specific correlations.

One way to remedy the problem with domain specificity could be to select query terms by measuring similarity, not to individual query terms, but to the entire query concept, as suggested by [10]. This means that we would first produce a query vector by, e.g., summing the context vectors of the words in the query, and then calculate similarity between the query vector and the context vectors of the words in the vocabulary. Qiu & Frei (1993) demonstrate that this methodology performs better on three standard test collections than traditional term based expansion.

4.1.3 Query typology

Queries are compact, but in their structure and in anticipated user needs we wil l be able to find types of anticipated retrieval. Some queries are to the point; others more vague. In a CLEF-type evaluation the query surface structure is of limited use since all have purposely been formulated similarly; in a real-life s etting variation will be more noticeable and give more purchase to typologization.

5 Conclusions

Using state-of-the-art morphological tools is necessary for compounding and inflecting languages such as the ones at hand. Utility of higher level linguistic analysis for information retrieval is yet unproven; in further experiments we plan to investigate the utility of clause-internal dependency relations for this purpose.

The baseline retrieval method must be improved, primarily by adding a mechanism for query term weighting. One possibility is to use information about typological features to derive different weighting schemes. We also noted that the retrieval function promote short documents over longer ones, i.e the normalization method should also be improved. Normalization is especially important in this domain since corpora of news articles usually contain a large amount of very short articles.

Co-occurrence-based query expansion gives patchy results when used without domain-specific and general linguistic guidance. Lexical resources or domain models would undoubtedly improve results. However, the major aim for our conceptual clustering experiments is to model human information processing — we wish to find data-driven models that can be trained on the data at hand, whatever it is, to discover and build such tools rather than make use of existing ones. This means we need a finer-grained model of co-occurrence, term distribution, and textual progression.

Acknowledgements The work reported here is partially funded by the European Commission under contracts IST-2000-29452 (DUMAS) and IST-2000-25310 (CLARITY) which is hereby gratefully acknowledged. We thank Tidningarnas TelegrambyråAB, Stockholm, for providing us with the Swedish text collection.

References

[1] R. Bayer and K. Unterauer. Prefix B-trees. *ACM Transactions on Database Systems*, 2(1):11–26, March 1977.

[2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[3] M. J. Folk, B. Zoellick, and G. Riccardi. *File Structures: An Object-Oriented Approach with C++*. Addison-Wesley, 3rd edition, 1998.

[4] Z. Harris. *Mathematical Structures of Language*. Interscience publishers, 1968.

[5] P. Kanerva, J. Kristofersson, and A. Holst. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, page 1036. Erlbaum, 2000.

[6] J. Karlgren and M. Sahlgren. From words to understanding. In Y. Uesaka, P. Kanerva, and H. Asoh, editors, *Foundations of Real World Intelligence*, pages 294–308. CSLI publications, 2001.

[7] S. Kaski. Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Proceedings of the IJCNN'98, International Joint Conference on Neural Networks*, pages 413–418. IEEE Service Center, 1998.

[8] T. Landauer and S. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[9] K. Lund and C. Burgess. Producing high-dimensional semantic spaces from lexical cooccurrence. *Behavior Research Methods, Instruments and Computers*, 28(2):203–208, 1996.

[10] Y. Qiu and H. P. Frei. Concepy based query expansion. In *Proceedings of the 16th ACM SIGIR conference on research and development in information retrieval*, pages 160–169, 1993.

[11] H. E. Williams and J. Zobel. Compressing integers for fast file access. *The Computer Journal*, 42(3):193–201, 1999.

[12] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing, 2nd edition, 1999.