Vector-based Semantic Analysis using Random Indexing and Morphological Analysis for Cross-Lingual Information Retrieval

Jussi Karlgren, Magnus Sahlgren
SICS, Stockholm
www.sics.se/humle/Homeosemy

Meaning in Information Access

Meaning, the arguably most important theoretical object of study in information access, is most decidedly situation-dependent. While much of meaning appears to achieve consistency across usage situations — a term will seem to mean much the same thing in many of its contexts — most everything *can* be negotiated on the go. Human processing appears to be flexible in this respect, and oriented towards learning from prototypes rather than learning by definition: learning new words, and adding new meanings or shades of meaning to an existing word does not need a formal re-training process. And, in fact, natural use of human languages does not make use of definitions or semantic delimitations; finding an explicit definition in natural discourse is a symptom of communicative malfunction, not of laudable explicitness.

We would like to build a text model which would invite processing in a human-oriented way. To do this we need to understand language better. But not any knowledge of language will do: a text model should model language *use* rather than language in the abstract. We need a better understanding of how meaning is negotiated in human language usage: fixed representations do not seem practical, and do not reflect observed human language usage. We need more exact study of inexact expression, of the *homeosemy* (homeo- from Greek *homoios* similar) or near and close synonymy of expressions of human language. This means we need to understand the temporality, saliency, and topicality of terms, relations, and grammatical elements – it means modeling the life cycle of terms in language, the life cycle of referents in discourse, and the connection between the two.

This is a tall order but the present experiments are inteded to be a first step in the direction of a flexible text model. Our hope is that if we can show that information retrieval experiments — especially cross-lingual ones — can benefit from associative and dynamic modelling of meaning and text, we are on the right track.

Overview

We have built a query expansion and translation tool. When used in one single language it will expand the terms of a query using a thesaurus built for that purpose; when used across languages it will provide numerous translations and near translations for the source language terms.

The underlying technology we are testing is that of vector-based semantic analysis, an analysis method related to latent semantic indexing based on stochastic pattern computing, described in other publications.

Our tool is built for people who want to formulate a query in another language, and is designed for interactive use. In this year's CLEF we have used it automatically with no human intervention to produce queries for crosslingual retrieval, but we have also designed a pleasing window-based interface for experimentation. The queries we produced were tested on a standard Inquery installation at another site.

Our approach, as a data-intensive method, relies on the availability of reasonably large amounts of relevant training data and on the adequate preprocessing of the training material.

This paper will briefly describe how we acquired training data, aligned it, analyzed it using morphological analysis tools, and finally built a thesaurus using the data, but will concentrate on an overview of vector-based semantic analysis and how stochastic pattern computing differs from latent semantic indexing in its current form.

Vector-Based Semantic Analysis using Random Indexing

Vector-based semantic analysis is a technology for extracting semantically similar terms from textual data by observing the distribution and collocation of terms in text. We have been experimenting with vector-based semantic analysis using stochastic patterns in a technique we call Random Indexing. The result of running a vector-based semantic analysis on a text collection is in effect a thesaurus: an associative model of term meaning. This can be used to build a synonym tool for application e.g. in query expansion. Similarly, the vector-based semantic analysis can be used to find correspondences across languages. If multi-lingual data are used, correspondences from them are as easy to establish as within a language.

Random Indexing uses sparse, high-dimensional random index vectors to represent documents (or context regions or textual unit of any size). Given that each document has been assigned a random index vector, term similarities can be calculated by computing a terms-by-contexts co-occurrence matrix. Each row in the matrix represents a term, and the term vectors are of the same dimensionality as are the random vectors assigned to documents. Each time a term is found in a document, that document's random index vector is added to the row for the term in question. In this way, terms are represented in the matrix by high-dimensional semantic context vectors which contain traces of each context the term has been observed in. The underlying assumption is that semantically similar terms will occur in similar contexts, and that their context vectors therefore will be similar to some extent. Thus, it should be possible to calculate the semantic similarity between any given terms by calculating the similarity between their context vectors (mathematically, this is done by calculating the cosine of the angles between the context vectors). This similarity measure will thus reflect the distributional (or contextual) similarity between terms.

This technique is akin to latent semantic analysis or indexing, except that no dimension reduction such as singular value decomposition is needed to reduce the dimensions of the co-occurrence matrix, since the dimensionality of the random index vectors is smaller than the number of documents in the text data. This makes the technique more efficient than the latent semantic indexing methods, since singular value decomposition is a computationally demanding operation: random index vectors of a dimension on the order of 1000 may be used to cover a wide range of vocabulary. The technique is also eminently more easily scalable and more flexible as regards unexpected data than are methods which rely on dimensional reduction: a new document does not require a larger matrix but will simply be assigned a new random index vector of the same dimensionality as the preceding ones and a new term requires no more than a new row in the matrix.

The size of the context used to accumulate the terms-by-contexts matrix may range from just a few adjacent words on each side of the focus word to entire documents consisting of more than hundred words. Both document-based co-occurrence statistics and narrow context windows have been used in experiments with Random Indexing to calculate semantic term vectors with favorable results.[1]

In the present experiment, we have used Random Indexing to index aligned bilingual corpora and extract semantically similar words across languages.

Training Data

For data we used a large number of documents of European legislation. The CLEF queries were used to retrieve documents from the WWW service provided by the commission in French, Swedish, and English. 77 documents of a few hundred sentences each were retrieved in each language.

The corpus used in these experiments consists of documents in several different languages downloaded from the Eur-Lex website (http://europa.eu.int/eur-lex/) by using keywords in the CLEF queries as search terms. It should be pointed out that, due to the somewhat narrow topical spread of the Eur-Lex database (which consists of legislation texts from the European Union), not every topic returned satisfying search results. Indeed, a few CLEF queries did not return one single relevant document, such as for example keywords to the query:

```
"C044: Indurain Wins Tour. Reactions to the fourth Tour de
France  won by Miguel Indurain. Relevant documents comment on
the reactions  to the fourth consecutive victory of Miguel
Indurain in the Tour de  France. Also relevant are documents
discussing the importance of  Indurain in world cycling after
this victory."
```

Each document was downloaded in each of the languages used in the experiments (English, French and Swedish). The documents were concatenated language by language to produce training corpora consisting of roughly 2 million

[1] For reports on experiments using these different context sizes, see Kanerva, P., Kristofersson, J. and Holst, A. (2000): Random Indexing of Text Samples for Latent Semantic Analysis. In Gleitman, L.R. and Josh, A.K. (Eds.). Proceedings of the 22nd Annual Conference of the Cognitive Science Society (p. 1036). Mahwah, New Jersey: Erlbaum; and Karlgren, J. and Sahlgren, M. (2001): From Words to Understanding. In Kanerva et al. Eds Real World Intelligence, forthcoming as a CSLI publication, Stanford: Center for the Study of Language and Information.

words in each language. Thus, the training data consisted of the same text translated into different languages. This amount of training data is probably near the absolute minimum to be able to provide any reasonable semblance of thesaurus functionality for our purposes; a topically uneven distribution will be immediately reflected in the consistency of results across topics.

Morphological Analysis

The texts were morphologically analyzed using tools from Conexor. Conexor's tools provided morphological base forms and, for Swedish, compound splitting. In future experiments, we intend to try make use of more text-oriented analyses as provided by syntactic components of the tools.

Alignment Algorithm

The translated texts were preprocessed by a series of Perl scripts aligned using a dynamic weighting algorithm written in SICStus Prolog. The algorithm gave high weight to headings and subheadings, anchoring the alignment, and then aligned sentences between anchor points using a matching function based on word overlap and relative length of clauses. This step was much aided by the fact that European legislation is written to match clause by clause between the various official languages: the alignment algorithm could assume that a perfect match was attainable. In fact, this principle is not always adhered to, and the alignment continues to be a non-trivial problem.

article premier	Match!	article 1
section 1	Match!	section 1
objectif de ensemble de la législation alimentaire général la présent proposition avoir notamment pour objectif de mettre en place une base global commun pour la législation alimentaire . elle établir de es principe commun régir la législation alimentaire , définir de es terme commun et créer un cadre général pour la législation alimentaire	?	overall aim of general food law one of the aim of this proposal be to provide a common comprehensive basis for food law
…	?	…
article 2	Match!	article 2

Unfortunately, processing constraints of various sorts proved problematic for the alignment algorithm. Only about half of the available data were actually processed through the aligner and this, after the data collection itself, proved the most crucial bottleneck for our query processing.

Thesaurus Tool for Cross Lingual Query Expansion

The first step in constructing a bilingual thesaurus is to assign a 1,000-dimensional sparse random index vector to each aligned document in the bilingual corpus. These 1,000-dimensional random index vectors consist of 6 randomly distributed -1s and +1s. An index vector thus has 3 randomly distributed -1s and 3 randomly distributed +1s, with the rest set to zero. The random index vectors are then used to accumulate one words-by-contexts co-occurrence matrix per language by adding a document's index vector to the row for a given word every time the word occurs in that document. Words are thus represented in the words-by-contexts matrices by 1,000-dimensional context vectors that represent the relative meaning of words.

The assumption is that, since the documents will (hopefully) be close translations of each other, they will (hopefully) consist of words that are close translations of each other. As the context vectors effectively consist of the sum of the index vectors of the documents that the words occur in, words that occur in similar documents - i.e. that "are about" similar things - will get similar context vectors. So by comparing the context vectors across languages - i.e. by calculating the cosine of the angles between the vectors - it is possible to extract for each word its nearest neighbors in the other language. Presumably, the nearest neighbor will be a translation, or a near translation, of the word in question.

So by extracting the five highest correlated terms for each word, we effectively produced a bilingual thesaurus. We also used a threshold for the correlations to avoid extracting terms with very low correlation to the focus word (since terms with low correlation are assumed to be semantically unrelated, or at least not comparatively similar). Such cases may appear for example when the focus word has a low frequency of occurrence in the training data. The threshold was set to exclude words with a correlation less than 0,2 (where 1 could be thought of as a complete match - i.e. two words that have occurred in exactly the same documents - and 0 a complete disaster - i.e. two words that have not co-occurred in a single document (although the randomness of the index vectors makes it possible for even distributionally unrelated words to get a correlation higher than 0)) to the target word.

Retrieval and CLEF results

The retrieval itself was done using an Inquery system set up by the Information Sciences department at Tampere University. [2]The finished queries were re-edited to Inquery query syntax and retrieved from the CLEF database. We submitted four runs to CLEF: three French-English bilingual runs: description (sicsfed), narrative (sicsfen), narrative+morphological analysis (sicsfenf), and one Swedish-English run: narrative+morphological analysis (sicssen). Using the narrative rather than the description gave about double the number of query terms (more than 20 terms per query rather than about ten); using the morphological analysis almost doubled the number of terms again. The Swedish run provided rather fewer query terms than did the French runs, owing to better aligned training data as far as we can determine.

The retrieval results by CLEF standards can fairly be characterized as underwhelming. For most queries in our submitted runs our results are well under the median.

Run name	Average precision	Relevant returned	Terms/query
SICSFED	0,1646	390	12
SICSFEN	0,2216	433	26
SICSFENF	0,0864	342	48
SICSSEN	0,0139	211	34

Interactive Interface and Web Service Availability

The thesaurus itself is quite demanding as regards memory and will naturally run on some server. It has been built with no regard to usability issues. For experimentation and demonstration purposes, we have designed and built

[2] We very gratefully thank Heikki Keskitalo for helping us gain access to the Inquery system used at his department.

Synkop, an interface for accessing the thesaurus tool over the WWW. Synkop, written entirely in Java using the Swing package, should be able to run on most platforms as a client to a thesaurus server. The thesaurus, as trained for CLEF purposes, will be made network accessible by us using some standard protocol in the near future. The example interface can be downloaded from our web page.

Lessons Learnt and Proposals for Future CLEFs

Our approach is based on constructing useful queries, not on more effective retrieval as retrieval. We used Inquery for testing our queries. A standard system for this purpose would be useful - and we suspect other groups might be interested in the same. Could we somehow organize a set-up where some site which experiments with retrieval systems sets up some such service over the net?

We have several query-oriented problems to work on for future years. This year we paid no attention to interaction between query terms, but translated them one by one. Next year we intend to address the likely collocations between synonym candidates to weed out unlikely combinations. In addition, we made little use of the language analysis tools at our disposal; the syntactic processing may well be crucial for improving understanding of context similarity and we plan to experiment with including syntactic information into the random indexing scheme during next year. Both query analysis and syntax should help us improve precision somewhat.

But recall is the major problem. Training data and its preprocessing are the major bottlenecks for our approach. Our results will vary with query topic and the availability of training data for that specific topic and domain. This year we only used one information source for the training data and used a an experimental and not very satisfactory alignment tool; for future years we intend to add sources and partially automate the training data acquisition process.

In a real-life retrieval situation the queries will have to be inspected by the person performing the search: we have built a demonstration interface to illustrate a likely retrieval process where someone searches data in a language they know but do not know well. We will most likely not attempt interactive experimentation, but the interface demonstration will serve as an example of what functionality we can provide. The main aim will continue to be to understand textuality and text understanding – information access is an excellent testing ground for the hypotheses we are working on.