Pruning Texts with NLP and Expanding Queries with an Ontology : TagSearch

Gil FRANCOPOULO
www.tagmatica.com

Abstract:

The basic line of our action is first to use natural language processing to prune the texts and the query, and secondly to use an ontology to expand the queries.

Last year

The system described here is based on the one used last year for CLEF-2002. But most components have been improved and some new steps have been added.

The system whose name is TagSearch is based on three main components:

- A chunker, named TagChunker[1].
- Lucene, a good OpenSource search engine written by Doug Cuting and his friends[2].
- An ontology, named TagDico[3].

The first two components were used last year. The use of an ontology is new.

Objectives

Our main objectives was to find the right documents in deducing implicit information and in avoiding noise.
The task is divided in two steps: index the texts and search in the index.

Main ideas for indexing

The idea is that instead of indexing characters strings, the texts are parsed and only the results of the parsing are indexed[4]. So we are able to prune the wrong parts of speech.
That means that is possible to:

a) To insert only the right part of speech inside the index. For instance, in the sentence : “the chair is there” the word “chair” is determined as a noun and not as a verb. So in the index, the pair “chair” + Noun is inserted. The goal is to avoid noise during searching.
b) To insert only the part of speech we want. We insert only the adjectives, nouns and verbs. Grammatical words and adverbs are not inserted, because we are not going to search against their meanings.
c) To use the word segmentation to group correctly compound words. And the segmentation is controlled by a French lexicon where a lot of compound words are described. For instance, if the compound word “pomme de terre” (potatoes in English) appears in a text, the whole string is inserted. In French, the words “pomme” and “terre” have nothing to do with “pomme de terre”, so the three words “pomme”, “terre” and “pomme de terre” must considered as being completely different words.
d) To correct the mistakes in the texts. TagChunker has a module called TagCorrector that is specialized in this task.

Main ideas for searching

The index being constructed, let’s see how we are going to evaluate the query against it.

First, we parse the query with the same chunker as for indexing. That means that, of course the part of speech tagging and word segmentation is exactly the same. The result is scanned and only the nouns, adjectives and verbs are retained. If a compound noun appears in the query it is recognized correctly.
Searching is done as follows, until giving 1000 documents:

Step-1: a Lucene query with an AND between the query words is automatically built and evaluated.
Step-2: all the words are expanded thru three types of links in the ontology: synonymy, meronymy[5] and derivation[6]. Each initial word of step-1 is grouped by a OR with its expansion. The various groups are still

[1] TALN-2003 (Batz-Sur-Mer) Francopoulo TagChunker : mécanisme de construction et évaluation.
[2] See: http://jakarta.apache.org/lucene/docs/index.html for details.
[3] See “www.tagmatica.com + Produits et services” for details
[4] Only the part of speech tagging is used. The chunker produces also chunk grouping and chunk labelling, but this information is not used.

connected by a AND. But instead of building one big query, the combination of each expansion is built, producing a lot of queries. For each query, the number of terms is computed and the evaluation starts with the queries that has the lesser terms.
Step-3: a query is built like in step-1, but instead of using AND, we use OR between words.
Step-4: queries are built like in step-2, but instead of using AND, we use OR between groups.

A few words on expansion

The goal of expansion is to find documents that are on the subject we search but without exactly the word we have in the query. For instance, in a query like "Les syndicats en Europe[7]" (query C156). Imaging a text in the pool of text that is about "syndicats en Italie" but without the word Europe. If the word "Europe" is not expanded, you cannot find this document. So meronymy expansion is the only possibility to find this document.

A few words on document ranking

The documents must be ranked. And a document that is found during step-1 must have a higher rank than the one computed during step-2 or 3.
I use the ranking produced by Lucene as a basis and I multiply this ranking by a number lesser that one in order to reflect the query ranking.

Results

As far as I know in reading the comparative results, our results seems to be not so bad. We are not in the worst results.

Future

The lexicon being rather rich, a problem occurs: in case of polysemy, the system does not prune the meanings that are described in the lexicon but that are not in the context of the sentence. That means that sometime the expansion is noisy.
The system could be improved by a semantic desambiguation component.

Conclusion

The query ranking has been adapted for CLEF-2003 but the rest (the chunker and the ontology) has been used as is.
The system works pretty well.
We did not have big problems to participate to the 2003 campaign.
It was easier than last year.

[5] "meronymy" is the relation "is part a of", for instance "Italy is a part of Europe".
[6] "derivation" is the relation that permits to link two related meanings with a different part of speech, for instance from a verb to a noun, by a link like "this is a name of the action" or "this is the name of the result of the action".
[7] Straightforward translation in English : « the syndicates in Europe ».