MediaLab @ CLEF-2002:
Comparing search strategies.

Dennis Reidsma, Peter van der Weerd
{d.reidsma, pweerd}@medialab.nl

MediaLab BV, Schellinkhout, The Netherlands
http://www.medialab.nl

Abstract

This report describes the participation of MediaLab BV in the CLEF-2002 evaluations. This year we participated in the monolingual Dutch task for the second time. Our main objective last year was to get some experience with participating in these experiments and to get a first impression on how our search engine performed compared to other search engines. This year we wanted to experiment with different search strategies and parameterisations on our core search engine.

1 Approach

All experiments were done using the core search engine developed by MediaLab.
We had two aims in mind: the first was to experiment with some completely different search strategies, the second was to run those strategies with variations on several parameters in order to investigate the effect on the quality of the results. The next section describes those strategies and parameters in detail. To compare the strategies we used the queries and assessments of last year, evaluating the results using the trec_eval program developed by NIST. Section 3 discusses those results, also comparing them to our performance last year. Section 4 is about our results with the topics for this year and section 5 presents our conclusions and some discussion on what we intend to do next year at CLEF-2003.

2 Strategies

In the working notes of the CLEF-2001 workshop we stated that we would extend our approach with Natural Language Processing and query word weighting for our participation in 2002 [2]. However, when we started preparing our submission for this year we decided that we should start with deciding on a good base line algorithm that uses only the core functionality of our search engine. For this we investigated two basic strategies, which are discussed in the rest of this section. The first strategy only searches for query words in a collection of simple indexes over the document fields, the second strategy uses co-occurrence indexing.

Common parts

The following is a list of parts that were common to both strategies.

- A stopword list was obtained by extending the CLEF stopword list for Dutch with our own default stopword collection.
- Stemming was done using a default algorithm based on the Porter stemmer.
- Compound terms were generated from the words in the freetext index: if for instance both “klap”, “roos” and “klaproos” are found in the index, “klaproos”is considered a compound term consisting of “roos” and “klap”. So, searching for “roos” will also retrieve hits on “klaproos”,

Strategy 1

The first strategy was very straightforward, since our aim was not to find the best algorithm but to define a base line algorithm that performs well and is easy to understand and analyse. We defined a collection of indexes over the document fields ‘TI’, ‘LE’ and ‘TE’ and combined indexes over every combination of those fields. The next step was to define runs that differed on the following points:

- Which indexes were searched (at most 3 indexes per run)
- Which parts of the query were used in searching (title, description and/or narrative). The different indexes in one run could be searched using a different combination of query fields.

- How strong the influence of keyword stemming should be for the separate indexes in a run. Since we wanted to keep the number of runs small, we used only the settings *no stemming*, *default stemming* (relative weight of 0.6), *full stemming* (same weight as exact matches).

Using these parameters, we defined about 500 runs on the topics of last year.

Strategy 2

With the second strategy we tried to improve the results of a run by using co-occurrence information. For this we first defined a co-occurrence network on the document collection. This network then was used to modify the query results in the following way:
For every word in the query it's co-occurrences were determined. Combining those co-occurrences for all words with an *AND* or an *OR* operation resulted in so called *strong query expansions* and *weak query expansions*. Those query expansions could then be used to find more query results or just rerank the results. The assumption was that this procedure might improve recall in the runs.

3 Results on last year's collection

All variations on both strategies were tested on the topics of last year. The result files that were produced were analysed using the trec_eval program. This section describes the outcome of that analysis.

Strategy 1

Table 1 shows a few of the best results we achieved using the first strategy. Given the fact that the average precision for the top 5 participants on that 2001 collection were 0.3917, 0.3844, 0.3775, 0.3497 and 0.2833 [1] it may be clear that the new strategy improved results drastically.
We did not have time to do a proper statistical analysis of the results to determine which values for the different parameters resulted in the best performance. Still, the results seem to indicate a trend that the best results are achieved using the title and description field of the query, leaving the narrative out and giving stemmed variants of the query words a relative weight of 0.6 (so 2 stemmed keyword matches are slightly more important than one exact match).

Run	**Relevant**	**Rel_ret**	**Average prec. (non-interp.)**	**R-precision**
279	1224	1006	0.3177	0.3287
87	1224	1007	0.3176	0.3351
71	1224	1017	0.3153	0.3225
Official 2001 submission	1224	879	0.1640	0.1803

Table 1: Retrieval results on 2001 collection

Strategy 2

The results produced with the second searching strategy were not as much improved. On the contrary, every variation on this strategy performed worse *with* than *without* using the co-occurrence enrichment. With hindsight it is not hard to think of an explanation why this 'had to be unavoidable', although we would have to devise some other experiments to verify that the explanation is the right one. Currently we are inclined to think that since co-occurrence information describes which words occur often together in a document, the co-occurrences of the query words are already bound to be in the top part of the results. That would mean that using this information to expand the results would not result in much improvement in recall, whereas the noise introduced by the expansion *would* result in the retrieval of more irrelevant results.

4 Our results this year

To produce our results for this year we used several variations on the first strategy that performed not too badly on last year's topics. We decided not to use the second strategy because it produced such poor results.

Though our best results on the topics of last year were pretty good, the comparisons to the median suggest that we did not perform as well on this year's topics. The difference is large enough to be more than somewhat surprising, but unfortunately we did not have time to find out what caused this behaviour.

5 Conclusions

Our achievements in improving the results for the topics of last year with such a simple strategy gives us confidence that we are on the right track. For next year we might experiment with several extensions to this base line algorithm, such as for example:

- Blind (negative) relevance feedback on the *lowest* retrieved results, such as was used last year by McNamee and Mayfielsd among others [3].
- Weighting the different query words based on word category, corpus frequency and language frequency
- More detailed experimenting with the optimal settings for stemming and compound word searching.

6 References

[1] *Working Notes for the CLEF 2001 Workshop*, edited by Carol Peters, September, Darmstadt, Germany

[2] *First experiments with CLEF* Peter van der Weerd, Wilfred Blom, Medialab BV, Schellinkhout, in [1]

[3] *APL Experiments at CLEF: Translation Resources and Score Normalization*, Paul McNamee and James Mayfield, Johns Hopkins University, USA, in [1]