The University of Amsterdam at CLEF 2001

Christof Monz Maarten de Rijke

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam, 1018 TV Amsterdam
The Netherlands
E-mail: {christof,mdr}@science.uva.nl

Abstract

This paper describes the official runs of our team for CLEF-2001. We took part in the monolingual task, for Dutch, German, and Italian. The focus of our experiments was on the effects of morphological analyses such as stemming and compound splitting on retrieval effectiveness. Confirming earlier reports on retrieval in compound splitting languages such as Dutch and German, we found improvements to be around 25% for German and as much as 55% for Dutch. For Italian, lexicon-based stemming resulted in gains of up to 25%.

1 Introduction

This is the first year that the University of Amsterdam is participating in the CLEF conference and competition. We took part in three monolingual tracks: Dutch, German, and Italian. We were particularly interested in the effects of shallow morphological analyses: stemming or lemmatization, and compound splitting. All experiments were performed using the FlexIR system.

The paper is organized as follows. In Section 2 we describe the FlexIR system as well as our approach. Section 3 describes our official runs for CLEF 2001, and in Section 4 we discuss the results we have obtained. Finally, in Section 5 we offer some conclusions regarding research within ILLC in the area of text retrieval.

2 System Description

All submitted runs used FlexIR, an information retrieval system developed by the first author. The main goal underlying FlexIR's design is to facilitate flexible experimentation with a wide variety of retrieval components and techniques. FlexIR is implemented in Perl; as it is built around the standard UNIX pipeline architecture, and supports many types of preprocessing, scoring, indexing, and retrieval tools.

2.1 Approach

The retrieval model underlying FlexIR is the standard vector space model. All our official runs for CLEF 2001 used the Lnu.ltc weighting scheme [2] to compute the similarity between a query (q) and a document (d):

$$sim(q,d) = \sum_{i \in q \cap d} \frac{\frac{1+\log(\text{freq}_{i,d})}{1+\log(\text{avg}_{j \in d}\text{freq}_{j,d})} \cdot \frac{\text{freq}_{i,q}}{\max_{j \in q}\text{freq}_{j,q}} \cdot \log\left(\frac{N}{n_i}\right)}{((1-slope)\cdot pivot + slope \cdot \text{unique\_words}_d) \cdot \sqrt{\sum_{i \in q}\left(\frac{\text{freq}_{i,q}}{\max_{j \in q}\text{freq}_{j,q}} \cdot \log\left(\frac{N}{n_i}\right)\right)^2}} \quad (1)$$

For the experiments on which we report in this note, we fixed *slope* at 0.2; the pivot was set to the average number of unique words occurring in the collection.

In addition, blind feedback was applied to expand the original query with related terms. Term weights were recomputed by using the standard Rocchio method [10], where we considered the top 10 documents to be relevant and the bottom 250 documents to be non-relevant. We allowed at most 20 terms to be added to the original query. We did not carry out any filtering [7] before applying Rocchio, since some experiments that we carried out on the CLEF 2000 data set indicated a decrease in retrieval effectiveness.

2.2 Inflectional Morphology

Previous retrieval experimentation [3] in English did not show consistent significant improvements by applying morphological normalization such as rule-based stemming [9] or lexical stemming [4].

As to the effect of stemming on retrieval performance for languages that are morphologically richer than English, such as Dutch, German, Italian or Spanish, a similar mixed picture from CLEF 2000 and other experiments. Kraaij and Pohlmann [5] report that for Dutch the effect if stemming is limited; it tends to help as many queries as it hurts. Likewise, for German and French, reports seem to indicate results similar to those for English [8].

In our participation in this year's edition of CLEF, we focused on Dutch, German and Italian. Although versions of Porter's stemmer are available for each of these languages, we decided to use a lexical-based stemmer, or lemmatizer, because it tends to be less aggressive than rule-based stemmers, and we conjectured that this might benefit further morphological analyses such as compound splitting (see below). The lemmatizer is part of the TreeTagger part-of-speech tagger [11]. Each word is assigned its syntactic root by lexical look-up. Mainly number, case, and tense information is removed, leaving other morphological processes such as nominalization intact. As an example in German, *Vereinbarung* (English: agreement) and German: *vereinbaren* (English: agree) are not conflated.

2.3 Compound Splitting

Compound splitting is not an issue in English since almost all compounds, such as *Computer Science*, *peace agreement*, etc. are separated by a white space, disregarding some exceptions such as *database* or *bookshelf*. In Dutch and German compounds are not separated and compound building is a very common phenomenon. Kraaij and Pohlman [6] show that compound splitting leads to significant improvement of retrieval performance for Dutch, and Moulinier et al. [8] obtain similar results for German.

In some of our official runs for Dutch and German we used a compound splitter. Our compound splitter for Dutch was built using the Dutch lexicon provided by Celex [1], while our German compound splitter used the part-of-speech information provided by TreeTagger. Although compounds can consist of words having different parts-of-speech, we limited our compound splitters to noun-noun compounds. For instance, hte German compound *Friedensvertrag* (English: *peace agreement*) is split into *Frieden*+s *Vertrag*. Each noun is analyzed recursively whether it can be seen as a sequence of concatenated nouns (allowing for a glueing-*s*).

For retrieval purposes, each document in the collection is analyzed and if a compound is identified, all of its parts are added to the document. In some cases, compound splitting can give rather awkward results, e.g., German: *Bahnhof* (English: train station) is split into *Bahn* (rail) and *Hof* (court/yard). Whereas 'rail' is semantically related to 'train station,' this is less obvious for 'court' or 'yard.' Hence, it can happen that compound splitting adds some rather unrelated words to a document causing a slight topic drift. The current version of our compound splitters are not tuned for retrieval purposes; for instance, we did not try to avoid the addition of unrelated compound parts.

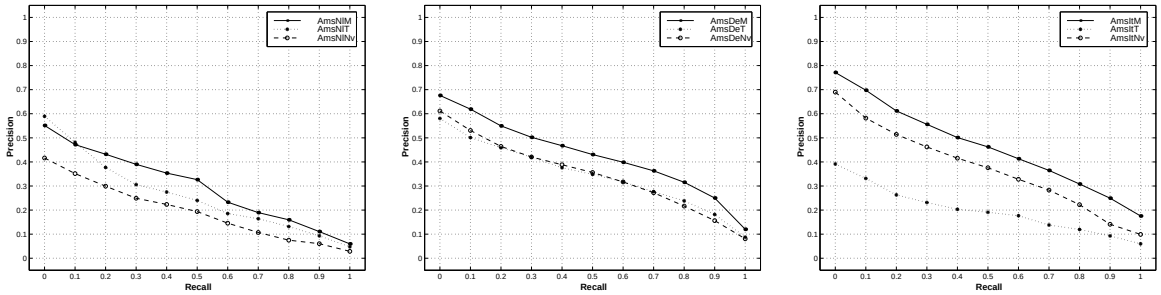
3 Runs

The University of Amsterdam participated in the monolingual task only, covering retrieval in Dutch, German, and Italian. For each language we submitted three types of runs:

Type M (Morphological) The title and the description field of the topic are used to generate the retrieval query (this was a mandatory requirement to be met by at least one of the runs). Words are morphologically normalized and compounds are split (Dutch and German). Blind feedback is applied to the top 10 documents adding at most 20 terms to the original query. This includes the runs `AmsNlM`, `AmsDeM`, and `AmsItM`.

Type Nv (Naïve) The title and the description field of the topic are used to generate the retrieval query. Blind feedback is applied to the top 10 documents adding at most 20 terms to the original query. In contrast to runs of type M, no morphological normalization or compound splitting are applied. This includes the runs `AmsNlNv`, `AmsDeNv`, and `AmsItNv`.

Type T (Title only) The same retrieval and document processing techniques are used as for runs of type M, but query formulation is restricted to the title field of the topic. This includes the runs `AmsNlT`, `AmsDeT`, and `AmsItT`.



Dutch German Italian

Figure 1: 11pt interpolated avg. precision for all submitted runs.

There are several motivations for this set of runs. Type M runs were intended to be the most effective runs, using techniques which are considered to improve retrieval effectiveness, such as blind feedback. Type T runs use the same techniques as type M runs, but queries are much shorter and, therefore, more closely resemble queries posed by a non-expert. Type Nv runs were intended as a contrast to type M runs, where no language specific techniques such as stemming/lemmatization or compound splitting are applied.

4 Results

This section summarizes some of the results of our CLEF 2001 submissions. Figure 1 displays the interpolated precision-recall curves for the three languages. Considering the non-interpolated avg. precisions for type M and type Nv runs in Table 1, one can see that morphological normalization does result in significant improvements[1] in effectiveness: $\approx 25\%$ for German and Italian and even $\approx 54\%$ for Dutch.

	Dutch	*German*	*Italian*
Naïve (Nv)	0.1833	0.3342	0.3580
+ Morphological Analysis (M)	0.2833 (+54.6%)	0.4172 (+24.8%)	0.4485 (+25.3%)

Table 1: Non-interpolated avg. precisions of Type M runs vs. Type Nv runs.

It is not obvious why the improvement for Dutch is so much bigger than for the other two languages. One reason could be that our precision scores for Dutch are, in general, considerably lower than the precision scores for German and Italian. Our results seems to suggest that the improvements brought about by compound splitting (plus stemming) is independent from the underlying retrieval engine.

	Dutch	*German*	*Italian*
Morphological Analysis (M)	0.2833	0.4172	0.4485
Title only (T)	0.2418 (-14.6%)	0.3342 (-19.9%)	0.1895 (-57.7%)

Table 2: Non-interpolated avg. precisions of TD-queries vs. T-queries.

Another interesting question is to compare queries that were formulated by using the title and the description field of the topic to queries that were formulated by using the title of the topic only. Queries based on title information only are much shorter and more closely resemble queries a non-expert would ask. Table 2 shows that for Dutch and German the decrease in effectiveness is certainly significant but not too dramatic. On the other hand, for Italian, using the title field only has a drastic impact on effectiveness, decreasing it by $\approx 57\%$. What causes this dramatic decrease, particularly in comparison to Dutch and German, is not obvious at this stage.

Finally, Table 3 shows the average precisions at a set of fixed ranks. This is again interesting from a regular user's point of view, who will hardly ever consider more than the top 20 documents returned by a retrieval system.

[1] Note that significant improvement here refers to the definition in [12], where changes of more than 5% are considered significant.

	Dutch			*German*			*Italian*		
	AmsNlM	AmsNlNv	AmsNlT	AmsDeM	AmsDeNv	AmsDeT	AmsItM	AmsItNv	AmsItT
p@5	0.3440	0.2760	0.3200	0.5102	0.4490	0.4286	0.5660	0.4255	0.2426
p@10	0.3080	0.2280	0.2480	0.5102	0.4163	0.4082	0.5170	0.4000	0.2106
p@15	0.2667	0.2027	0.2173	0.4721	0.4122	0.3878	0.4638	0.3645	0.1957
p@20	0.2500	0.1840	0.1970	0.4582	0.3878	0.3582	0.4330	0.3340	0.1766
p@30	0.2173	0.1667	0.1733	0.4102	0.3503	0.3218	0.3695	0.3007	0.1539
p@100	0.1194	0.0890	0.0942	0.2504	0.2143	0.1980	0.1970	0.1572	0.0898
p@200	0.0767	0.0552	0.0574	0.1669	0.1441	0.1288	0.1147	0.0951	0.0618
p@500	0.0367	0.0264	0.0306	0.0793	0.0715	0.0669	0.0502	0.0457	0.0338
p@1000	0.0193	0.0140	0.0195	0.0410	0.0380	0.0381	0.0255	0.0240	0.0202

Table 3: Avg. precision at rank *n*.

5 Conclusions

The experiments carried out here strongly confirm the believe that morphological normalization does improve retrieval effectiveness significantly. Since the morphological analyses carried out in this paper were still rather restricted, it would be interesting to see what impact additional analyses, e.g., stripping off prefixes and recognizing nominalizations, would have. Another line of interesting questions concerns the relation between the topic drift and the addition of parts of compounds.

Acknowledgments

Christof Monz was supported by the Physical Sciences Council with financial support from the Netherlands Organization for Scientific Research (NWO), project 612-13-001. Maarten de Rijke was supported by the Spinoza project 'Logic in Action' and by grants from the Netherlands Organization for Scientific Research (NWO), under project numbers 612-13-001, 365-20-005, 612.069.006, 612.000.106, and 220-80-001.

References

[1] R. Baayen, R. Piepenbrock, and L. Gulikers. The CELEX lexical database (release 2). Distributed by the Linguistic Data Consortium, University of Pennsylvania, 1995.

[2] C. Buckley, A. Singhal, and M. Mitra. New retrieval approaches using SMART: TREC 4. In D. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 25–48. NIST Special Publication 500-236, 1995.

[3] W. Frakes. Stemming algorithms. In W. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Strcutures & Algorithms*, pages 131–160. Prentice Hall, 1992.

[4] D. Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42:7–15, 1991.

[5] W. Kraaij and R. Pohlmann. Viewing stemming as recall enhancement. In *Proceedings SIGIR'96*, pages 40–48, 1996.

[6] W. Kraaij and R. Pohlmann. Comparing the effect of syntactic vs. statistical phrase index strategies for Dutch. In *Proceedings ECDL'98*, pages 605–617, 1998.

[7] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 206–214, 1998.

[8] I. Moulinier, J. McCulloh, and E. Lund. West Group at 2001: Non-English monolingual retrieval. In *Proceedings CLEF-2000*, 2000.

[9] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[10] J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System — Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[11] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, 1994.

[12] K. Sparck Jones. Automatic indexing. *Journal of Documentation*, 30(4):393–432, 1974.