PLIERS AND SNOWBALL AT CLEF 2002

A MacFarlane,
Centre for Interactive Systems Research, Department of Information Science,
City University, Northampton Square, LONDON EC1V 0HB, UK

Abstract: We test the utility of European language stemmers created using the Snowball language [1]. This allows us to experiment with PLIERS in languages other than English. We also report on some BM25 tuning constant experiments conducted in order to find the best settings for our searches.

1. Introduction

In this paper we briefly describe our experiments at CLEF 2002. We address a number of issues as follows. The snowball language was recently created by Porter [2] in order to provide generic mechanism for creating stemmers. The main purpose of the experiments was to investigate the utility of these stemmers and as to whether a reasonable level of retrieval effectiveness could be achieved by using Snowball in an information retrieval system. This allowed us to port the PLIERS system [3] for use on languages other than English. We also investigate the variation of tuning constants for the BM25 weighting function for the chosen European languages. The languages used for our experiments are as follows: German, French, Dutch, Italian, German and Finnish. All experiments were conducted on a Pentium 4 machine with 256 MB of memory and 240 GB of disk space. The operating system used was Red Hat Linux 7.2. All search runs were done using the Robertson/Sparck Jones Probabilistic model – the BM25 weighting model was used. All our runs are in the monolingual track. All queries derived from topics are automatic.

The paper is organised as follows. Section two describes our motivation for doing this research. In section three we describe our indexing methodology and results. In section 4 we describe some preparatory results using CLEF 2001 data. Section 5 describes our CLEF 2002 results, and a conclusion is given at the end.

2. Motivation for the Research

We have several different strands in our research agenda. The most significant of these is the issue of using Snowball stemmers in information retrieval, both in terms of retrieval effectiveness and retrieval efficiency. In terms of retrieval efficiency we want to quantify the cost of stemming both for indexing collections and servicing queries over them. How expensive is stemming in terms of time when processing words? Our hypothesis for search would be that stemming will increase inverted list size and that this would in turn lead to slower response times for queries. Stemming will slow down indexing, but by how much? Due to time constraints we restrict our discussion on search efficiency to CLEF 2002 runs. With respect to retrieval effectiveness we hope to demonstrate that using Snowball stemmers leads to an increase in retrieval effectiveness for all languages, and any deterioration in results should not be significantly worse. Our hypothesis is therefore: “stemming using Snowball will lead to an increase in retrieval effectiveness”. A further issue that we wish to address is that of tuning constants for the BM25 weighting model. Our previous research on this issue has been done on collections of English [3], but we want to investigate the issue on other types of European languages. A hypothesis is formulated in section 4 where the issue is discussed in more detail.

3. Indexing methodology and results

3.1 Indexing methodology

We used a simple and straightforward methodology for indexing: parsing, remove stop words, stemming in the given language. The PLIERS HTML/SGML parser needed to be altered to detect non-ascii characters such as those with umlauts, accents, circumflexes etc. The stemmers were easily incorporated into the PLIERS library. We used various stop word lists for the language, gathered from the internet. The official runs for Finnish did not use stemming, as no stemmer was available for that language while experiments were being conducted.

3.2 Indexing results

Language	Elapsed Time (mins)	Dictionary file size MB	Postings file size MB	Map file size MB	% of text
German	34.3	22.58	149.83	7.29	34%
French	23.6	4.36	48.16	3.13	23%
Dutch	36.1	12.49	138.98	6.00	30%
Italian	33.2	6.04	62.37	3.49	26%
Spanish	44.3	8.14	135.46	7.35	30%
Finnish	10.2	21.07	37.17	1.91	45%

Table 1 – Indexing results for builds using stemmers

The indexing results are given in tables 1 and 2. We report the following information: elapsed time in minutes, dictionary file size in MB, postings file size in MB, data file size in MB, and the size of the build files compared with the text file. The dictionary file contains keywords detected in the indexing process together with a link to the inverted lists contained in the postings file, while the data (or map) file contains information such as the CLEF identifier and location on disk.

Language	Elapsed Time (mins)	Dictionary file size MB	Postings file size MB	Map file size MB	% of text
German	8.28	28.6	159.3	7.29	37%
French	3.28	5.93	67.3	3.13	32%
Dutch	7.72	15.6	145.1	6.00	32%
Italian	4.05	8.09	87.2	3.49	36%
Spanish	6.78	9.95	140.6	7.35	31%
Finnish	2.30	21.1	37.2	1.91	45%

Table 2 – Indexing results for builds without using stemmers

The key results here are that stemmed builds take up slightly less space for most languages (it is a significant saving for French and Italian) and that builds with stemming take significantly longer than builds with no stemming. Builds with no stemming index text at a rate of 3.7 to 4.5 GB per hour compared with 0.48 to 0.89 GB per hour for stemmed builds. The results for stemmed builds are acceptable however.

4 Preparatory experiments: working with CLEF 2001 data

In order to find the best tuning constants for our CLEF 2002 runs we conducted tuning constant variation experiments on the CLEF 2001 data for the following languages: French, German, Dutch, Spanish and Italian. We were unable to conduct experiments with Finnish data as this track was not run in 2001: we arbitrarily chose K1=1.5 and B=0.8 for our Finnish runs.

We give a brief description of the BM25 tuning constants being discussed here [4]. The K1 constant alters the influence of term frequency in the BM25 function, while the B constant alters the influence of normalised average document length. Values of K1 can range from 0 to infinity, whereas the values of B are with the range 1 (document lengths used unaltered) to 0 (document length data not used at all).

We used the following strategy for tuning constant variation. For K1 we start with a value of 0.25 with increments of 0.25 to a maximum of 3.0, stopping when it was obvious that no further useful data could be gathered. For B we used a range of 0.1 to 0.9 with increments of 0.1. A maximum of 135 experiments were therefore conducted for each language.

Table 3 show the best tuning constants found for the CLEF 2002 together with the query type used for that tuning constants combination [Note: T = Title only queries, TD=Title and Description]. Note that these tuning constants were found by running experiments on the stemming builds, for application on runs of both type of build.

Language	K1 Constant	B Constant	Query Type
German	1.75	0.5	T and TD
French	2.0	0.5	T and TD
Dutch	2.75	0.8	T and TD
Italian	2.5	0.6	TD
Spanish	2.75	0.5	TD
Spanish	1.75	0.6	T

Table 3 – CLEF 2001 experiment results with chosen tuning constants (builds with stemming)

Tables 4 and 5 show the best results using CLEF 2001 data: using the tuning constants declared in Table 3. We formulate the following hypothesis for tuning constants on the CLEF collections: "the best tuning constants are independent of a given query set". In other words the best tuning constants found in our CLEF 2001 experiments should also be the best for our CLEF 2002 experiments.

Language	Average precision	Precision @5	Precision @10	Query Type
German	.213	.363	.326	T
	.269	.392	.373	TD
French	.243	.253	.226	T
	.254	.302	.265	TD
Dutch	.207	.268	.208	T
	.242	.336	.238	TD
Italian	.242	.260	.260	T
	.253	.306	.289	TD
Spanish	.210	.286	.273	T
	.225	.375	.322	TD

Table 4 – CLEF 2001 experiment results on builds without stemming

Language	Average precision	Precision @5	Precision @10	Query Type
German	.192	.335	.300	T
	.207	.367	.342	TD
French	.238	.269	.247	T
	.256	.286	.249	TD
Dutch	.193	.236	.208	T
	.228	.360	.292	TD
Italian	.266	.315	.300	T
	.305	.379	.336	TD
Spanish	.264	.343	.322	T
	.255	.427	.365	TD

Table 5 – CLEF 2001 experiment results on builds with stemming

Due to lack of time, our aim in these experiments was to achieve a better than baseline retrieval effectiveness for our preparatory CLEF 2001 experiments. For the most part we succeeded in doing this for both types of build. However we can separate our results into three main groups:

- For Dutch and German we were able to better six systems with our runs.
- For French and Italian, we were unable to better more than one run, but our effectiveness is considerably better than the official baseline runs.
- For Spanish we were unable to show much of an improvement over the baseline run.

Having said that we have a long way to go before our runs achieve the levels of performance of groups such as the University of Neuchatel particularly for languages such as French and Spanish.

We were unable to investigate the reason for the levels of performance achieved because of time constraints, but it is believed that the automatic query generator used to select terms for queries is simplistic and needs to be replaced with a more sophisticated mechanism. Our reason for believing that this might be the problem is that our results for Title only queries on the Spanish run are superior to those queries that were derived from Title and Description: this is counter to what we would expect.

When comparing experiments on those builds which used stemming and those that did not, we can separate our runs into three main groups:

- Stemming is an advantage: We were able to demonstrate that stemming was a positive advantage for both Italian and Spanish runs.
- Stemming makes no difference: Using stemming on French made very little difference either way.
- Stemming is a disadvantage: Stemming proved to be problematic for both Dutch and German.

We need to investigate the reason for these results. We are surprised that stemming is a disadvantage for any language

5 CLEF 2002 results

5.1 Retrieval efficiency results

Language	Average elapsed Time (secs)	Average Query Size	Query Type	Run Identifier
German	0.97	16.1	TD	plge02td
	0.22	3.84	T	plge02t
French	0.05	10.2	TD	plfr02td
Dutch	1.51	34.5	TD	ptdu02td
	0.28	4.50	T	ptdu02t
Italian	0.33	12.0	TD	plit02td
Spanish	1.23	17.2	TD	plsp02td
	0.52	5.06	T	plsp02t
Finnish	0.12	13.6	TD	plfn02td
	0.01	12.2	T	plfn02t

Table 6 – CLEF 2002 experiment results with chosen tuning constants (official runs)

Table 6 shows timings for each official run together with the average sizes of each query. All our runs have met the one to ten second response time criteria specified by Frakes [5], and all bar one have sub second response times. Title description runs are significantly slower than title only.

Language	Average elapsed Time (secs)	Average Query Size	Query Type
German	1.07	16.6	TD
	0.19	3.82	T
French	0.05	10.4	TD
Dutch	1.47	34.9	TD
	0.03	2.80	T
Italian	0.28	12.1	TD
Spanish	1.09	16.9	TD
	0.47	4.90	T

Finnish	0.11 0.01	13.2 3.50	TD T

Table 7 – CLEF 2002 experiment results with chosen tuning constants (alternative runs)

Table 7 show the alternative runs which means that searches were conducted on builds with no stemming, apart from Finnish where we did have a stemmer available. As with our official runs the response times are acceptable. With respect to the difference in response time between the two types of build, we can separate the results into three main groups:

- Runs on Stemmed builds are faster: German (TD).
- Runs on No-Stemmed builds are faster: German (T), Dutch, Italian, and Spanish.
- No significant difference in response times: French, Finnish.

In general this is what we would expect as inverted lists on stemmed builds tend to be larger than those of builds with no stemming. It is interesting to examine the exceptions and outliers, however. The reason German (TD) runs are faster on stemmed builds, is that the average query size is slightly larger by about half a term (more inverted lists are being processed on average). The Dutch no stem run is significantly faster on average than those of stemmed runs, but this is largely due to query size: queries with no stems on the Dutch collection are have 1.7 less terms on average than those of stemmed queries (fewer inverted lists are being processed). An interesting result with title only Finnish runs is that queries with no stems are nearly 3.5 times smaller than stemmed queries, but runs times are virtually identical: execution speeds are so small here it is difficult to separate them. It should be noted when we compared the size of queries with stems to those without stems, there is no clear pattern. We would suggest therefore that a simple hypothesis which suggested that runs on builds without stemming is faster on average than runs on stemmed builds cannot be supported with the evidence given here. It is clear that the number of inverted lists processes is an important factor as well as the size of the inverted lists.

Language	**Average precision**	**Precision @5**	**Precision @10**	**Query Type**
German	0.147 0.173	0.228 0.284	0.210 0.254	T TD
French	0.248	0.340	0.298	TD
Dutch	0.193 0.259	0.308 0.396	0.282 0.346	T TD
Italian	0.266	0.384	0.318	TD
Spanish	0.217 0.255	0.320 0.376	0.294 0.354	T TD
Finnish	0.123 0.171	0.200 0.240	0.143 0.190	T TD

Table 8 – CLEF 2002 official runs

5.2 Retrieval effectiveness results

Table 8 shows our official results for CLEF 2002. In general the results are quite disappointing and much lower in terms of average precision than our CLEF 2001 runs. We believe that our results are on the low side compared with other participants.

Language	Average precision	Precision @5	Precision @10	Query Type
German	0.159	0.292	0.253	T
	0.240	0.404	0.344	TD
French	0.237	0.271	0.244	TD
Dutch	0.246	0.384	0.334	T
	0.315	0.456	0.410	TD
Italian	0.198	0.282	0.239	TD
Spanish	0.235	0.388	0.344	T
	0.268	0.432	0.382	TD
Finnish	0.078	0.147	0.133	T*
	0.074	0.160	0.140	T
	0.069	0.147	0.123	TD

Table 9 – CLEF 2002 Comparative runs (* run on first stemmer attempt)

Table 9 shows the results on comparative runs, all using builds without stemming apart from Finnish where we did have a stemmer available. When comparing runs on different types of build we can separate our results into two main groups:

- Stemming is an advantage: French and Italian.
- Stemming is a disadvantage: German, Dutch, Spanish and Finnish.

The status for German and Dutch is unchanged from our CLEF 2001 results (stemming runs produced worse results), and also for Italian where the stemmer runs produced better results. Our results for French have improved comparatively, but for Spanish the results have deteriorated. The runs on the Finnish collection are particularly disappointing: the results for average precision on builds with stemming being about 40% worse for title only queries and nearly 60% worse on title/description queries than experiments on builds without stemming. The reason for this loss in performance could be because of the morphological complexity of Finnish and merits significant further investigation. It is also interesting that the initial version of the Finnish stemmer did slightly better in terms of average precision than the final version: this is offset with a slight loss in precision at 5 and 10 documents retrieved. The reduced effectiveness found on title/description queries compared with title only queries in our Spanish CLEF 2001 experiments was not repeated in our CLEF 2002 runs. However there is a slight loss in performance on the second version of the Finnish stemmer when comparing title/description queries to title only queries: this loss is consistent across all shown precision measures.

6 Conclusion

A number of research questions have been raised during this investigation of the effectiveness of stemming utilitising Snowball. They are as follows:

- Why are results on Dutch and German consistently worse using Snowball stemmers?
- Why are the results using the Finnish snowball stemmer significantly worse?
- Why are the results on the Spanish collection inconsistent, both with the Snowball stemmer and varying the query type (title only and title/description)?
- Why are the runs on Italian collections consistent, and how do we use evidence from these runs to improve the results of other romance languages such as Spanish and French?

Our hypothesis that suggested that the use of Snowball stemmers is always beneficial has not been confirmed. It may be possible to investigate this hypothesis further when we have addressed the research questions given above. We also have some conclusions with regard to retrieval efficiency and stemming:

- Stemming using the Snowball stemmers is costly when indexing, but does not slow down the process of inverted file generation to an unacceptable level.
- We have confirmed that inverted file size is not the only factor in search speed, query size that requires processing of more inverted lists play an important part as well.

We are unable to comment on our tuning constant experiments due to time constraints, and hence our hypothesis for this workshop version of the paper, but will have a poster which shows the results graphically and the information will be included in the final version of the paper.

Acknowledgements

The author is grateful to Martin Porter for his efforts to produce a snowball stemmer for Finnish.

References

[1] Snowball web site [http://snowball.sourceforge.net] – visited 19th July 2002.

[2] Porter. M., Snowball: A language for stemming algorithms,
[http://snowball.sourceforge.net/texts/introduction.html] – visited 19th July 2002.

[3] MacFarlane, A., Robertson, S.E.., McCann, J.A., PLIERS AT TREC8, In: Voorhees, E.M., and Harman, D.K., (eds), The Eighth Text Retrieval Conference (TREC-8), NIST Special Publication 500-246, NIST: Gaithersburg, 2000, p241-252.

[4] Robertson, S.E., and Sparck Jones, K., Simple, proven approaches to text retrieval, University of Cambridge Technical report, May 1997, TR356 ,
[http://www.cl.cam.ac.uk/Research/Reports/TR356-ksj-approaches-to-text-retrieval.html] – visited 22nd July 2002.

[5] Frakes, W.B., Introduction to information storage and retrieval systems. In: Frakes, W.B. and Baeza-Yates, R. (eds), Information retrieval; data structures and algorithms, Prentice Hall, 1992, p1-12.