SINAI at CLEF Ad-Hoc Robust Track 2007: applying Google search engine for robust cross-lingual retrieval

Fernando Martínez-Santiago, Arturo Montejo-Ráez, Miguel A. García-Cumbreras
Department of Computer Science. University of Jaén, Jaén, Spain
{dofer,amontejo,magc}@ujaen.es

Abstract

We have reported on our experimentation for the Ad-Hoc Robust track CLEF task concerning web-based query generation for English and French collections. We have continued the approach of the last year, although the model has been modified. Last year we used Google in order to expand the original query. This year we don't expand the query but we rather make a new query to be executed. Thus, we have to deal with two lists of relevant documents, one from each query. In order to integrate both lists of documents we have applied logistic regression merging solution. Obtained results are discouraging.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries

General Terms

Algorithms, Languages, Performance, Experimentation

Keywords

Information Retrieval, Multilingual Information Retrieval, Robust Information Retrieval

1 Introduction

Expanding user queries by using web search engines such as Google has been successfully used for improving the robustness of retrieval systems over collections in English[2]. Due to the multilinguality of the web, we have assumed that this could be extended to additional languages, though the smaller amount of web non-English pages could be a major counterpoint. Therefore, we have used Google in order to expand the query in a similar way [3], but instead of replacing the original query by the the expanded query, we have executed both queries (the original and expanded one). For each query we have obtained a list of relevant documents. Thus, we need to combine the retrieval results from these two independent list of documents. This is a similar problem to the so called collection fusion problem [8], but we have not several collections: there is only a collection but two list of relevant documents. The question is how should we make the calculus of the score of each document in the final resulting list. Given a query, in order to integrate the information available about the relevance of every retrieved document, we have applied a model based on logistic regression. Logistic regression has been used successfully in multilingual scenarios[5, 7].

2 Query expansion with Google search engine

This section describes the process for generating a new query using expansion by the Google search engine. To this end, we have selected a random sample document. The following fields correspond to the document with identification number `10.2452/252-ah` from the English collection.

```
<title>pension schemes in europe </title>

<desc>find documents that give information about current pension
systems and retirement benefits in any european country. </desc>

<narr>relevant documents will contain information on current pension
schemes and benefits in single european states. information of
interest includes minimum and maximum ages for retirement and the way
in which the retirement income is calculated. plans for future
pension reform are not relevant. </narr>
```

These fields have been concatenated into one single text and all contained nouns, noun phrases and prepositional phrases have been extracted by means of *TreeTagger*. TreeTagger is a tool for annotating text with part-of-speech and lemma information which has been developed at the Institute for Computational Linguistics of the University of Stuttgart[1].

Once nouns and phrases are identified they are taken to compose the query, preserving phrases thanks to Google's query syntax.

```
documents ''pension schemes'' benefits retirement information
```

The former string is passed to Google and the snippets (small fragment of text from the associated web page result) of the top 100 results are joined into one single text wherefrom, again, phrases are extracted with their frequencies to generate a final expanded query. The 20 most frequent nouns, noun phrases and prepositional phrases from this generated text are replicated according to their frequencies in the snippets-based text and then normalized to the minimal frequency in those 20 items (i.e. normalized according to the least frequent phrase among the top ones). The resulting query is shown below:

```
pension pension pension pension pension pension pension pension
pension pension pension pension pension pension pension pension
pension pension pension benefits benefits benefits benefits benefits
benefits benefits benefits benefits benefits retirement retirement
retirement retirement retirement retirement retirement retirement
retirement retirement retirement age age pensions occupational
occupational occupational occupational schemes schemes schemes
schemes schemes schemes schemes schemes schemes schemes schemes
schemes schemes schemes schemes schemes regulations information
information information information information scheme scheme
disclosure disclosure pension schemes pension schemes pension
schemes pension schemes pension schemes pension schemes pension
schemes pension schemes pension schemes pension schemes pension
schemes pension schemes retirement benefits schemes members members
occupational pension schemes occupational pension schemes
occupational pension schemes retirement benefits retirement benefits
disclosure of information
```

French documents have been processed in a similar way, but using the OR operator to join found phrases for the generated Google query. This has been done due to the smaller number of indexed web pages in French language. Since we expect to recover 100 snippets, we have found that with this operator this is possible, despite low quality texts been considered to produce the final expanded query.

The next step is to execute both original and Google queries on the Lemur information retrieval system. The collection dataset has been indexed using Lemur IR system[2]. It is a toolkit that supports indexing of large-scale text databases, the construction of simple language models for documents, queries, or subcollections, and the implementation of retrieval systems based on language models as well as a variety of other retrieval models. The toolkit is being developed

[1] Available at `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`

[2] http://www.lemurproject.org/

as part of the Lemur Project, a collaboration between the Computer Science Department at the University of Massachusetts and the School of Computer Science at Carnegie Mellon University. In these experiments we have used Okapi as weighting function ([4]).

Finally, we have to merge both lists of relevant documents. [1, 6] propose a merging approach based on logistic regression. Logistic regression is a statistical methodology for predicting the probability of a binary outcome variable according to a set of independent explanatory variables. The probability of relevance to the corresponding document D_i will be estimated according to four parameters: the score and the ranking obtained by using the original query, and the score and the ranking by means of the Google-based query (see equation 1). Based on these estimated probabilities of relevance, the list of documents will be interleaved making up an unique final list.

$$Prob[D_i \; is \; rel | rank_{org_i}, rsv_{org_i}, rank_{google_i}, rsv_{google_i}] = \frac{e^{\alpha+\beta_1 \cdot \ln(rank_{org_i})+\beta_2 \cdot rsv_{org_i}+\beta_3 \cdot \ln(rank_{google_i})+\beta_4 \cdot rsv_{google_i}}}{1+e^{\alpha+\beta_1 \cdot \ln(rank_{org_i})+\beta_2 \cdot rsv_{org_i}+\beta_3 \cdot \ln(rank_{google_i})+\beta_4 \cdot rsv_{google_i}}} \quad (1)$$

The coefficients α, β_1, β_2, β_3 and β_4 are unknown parameters of the model. When fitting the model, usual methods to estime these parameters are maximum likelihood or iteratively re-weighted least squares methods.

As it is needed to fit the underlying model, training set (topics and their relevance assessments) must be available for each monolingual collection. Since there are relevance assessments for English and French, we have made the experiments for these languages only. For Portuguese we have reported only the base case (we have not used Google queries for such language).

3 Results

As tables 1 and 2 show, the results are disappointing. For training data, Google queries improve both the m.a.p. and the geometric precision in both languages, English and French. But this good behavior disappears when we apply our approach on test data. Of course, we hope that precision for test data gets worse regarding training data, but we think that the difference in precision is excessive. This issue demands further analysis by us.

Approach	**Collection**	**map**	**gm-ap**
Google	training	0.29	0.12
Base	training	0.26	0.10
Google	test	0.34	0.12
Base	test	0.38	0.14

Table 1: Results for English data. Google approach is the result obtained by merging original queries and Google queries. Base results are those obtained by means of original queries only.

Approach	**Collection**	**map**	**gm-ap**
Google	training	0.28	0.10
Base	training	0.26	0.12
Google	test	0.30	0.11
Base	test	0.31	0.13

Table 2: Results for French data. Google approach is the result obtained by merging original queries and Google queries. Base results are those obtained by means of original queries only.

4 Conclusions and Future work

We have reported on our experimentation for the Ad-Hoc Robust Multilingual track CLEF task involving web-based query generation for English and French collections. The generation of a final list of results by merging search results obtained from two different queries has been studied. These two queries are the original one and a new one generated from Google results. Both lists are joined by means of logistic regression, instead of using an expanded query as we did last year. The results are disappointing. While results for training data are very promising, there is not improvement for test data. This question must be find out and we hope to understand why the performance is so poor for test data, analyzing, for instance, side effects of the regression approach.

5 Acknowledgments

This work has been partially supported by a grant from the Spanish Government, project TIMOM (TIN2006-15265-C06-03), and the RFC/PP2006/Id_514 granted by the University of Jaén..

References

[1] A. Calvé and J. Savoy. Database merging strategy based on logistic regression. *Information Processing & Management*, 36:341–359, 2000.

[2] K. L. Kwok, L. Grunfeld, and D. D. Lewis. TREC-3 ad-hoc, routing retrieval and thresholding experiments using PIRCS. In *Proceedings of TREC'3*, volume 500, pages 247–255, Gaithersburg, 1995. NIST.

[3] Fernando Martínez-Santiago, Arturo Montejo-Ráez, Miguel A. García-Cumbreras, and L. Alfonso Ureña-López . SINAI at CLEF 2006 Ad Hoc Robust Multilingual Track: Query Expansion using the Google Search Engine. *Evaluation of Multilingual and Multi-modal Information Retrieval 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006. LNCS-Springer.*, 4730, September 2007.

[4] S.E. Robertson and S.Walker. Okapi-Keenbow at TREC-8. In *Proceedings of the 8th Text Retrieval Conference TREC-8, NIST Special Publication 500-246*, pages 151–162, 1999.

[5] Fernando Martínez Santiago, Luis Alfonso Ureña López, and Maite Teresa Martín-Valdivia. A merging strategy proposal: The 2-step retrieval status value method. *Inf. Retr.*, 9(1):71–93, 2006.

[6] J. Savoy. Cross-Language information retrieval: experiments based on CLEF 2000 corpora. *Information Processing & Management*, 39:75–115, 2003.

[7] J. Savoy. Combining multiple strategies for effective cross-language retrieval. *Information Retrieval*, 7(1-2):121–148, 2004.

[8] E. Voorhees, N. K. Gupta, and B. Johnson-Laird. The collection fusion problem. In D. K. Harman, editor, *Proceedings of the 3th Text Retrieval Conference TREC-3*, volume 500-225, pages 95–104, Gaithersburg, 1995. National Institute of Standards and Technology, Special Publication.