Crosslanguage Retrieval based on Wikipedia Statistics

Andreas Juffinger, Roman Kern and Michael Granitzer

Know-Center, Graz
ajuffinger,rkern,mgranitzer@know-center.at

Abstract. In this paper we present the methodology, implementations and evaluation results of the crosslanguage retrieval system we have developed for the Robust WSD Task at CLEF 2008. Our system is based on query preprocessing for translation and homogenisation of queries. The presented preprocessing of queries includes two stages: Firstly, a query translation step based on term statistics of cooccuring articles in Wikipedia. Secondly, different disjunct query composition techniques to search in the CLEF corpus. We apply the same preprocessing steps for the monolingual as well as the crosslingual task and thereby acting fair across these tasks. The evaluation results reveal that the fairness comes at nearly no costs for monolingual retrieval but enables us to do crosslanguage retrieval and a feasible comparison of the performance on these two tasks.

1 Introduction

The goal of the task was to test whether WSD can be used beneficially for retrieval systems [2]. The organisers believe that polysemy is among the reasons for information retrieval systems to fail.

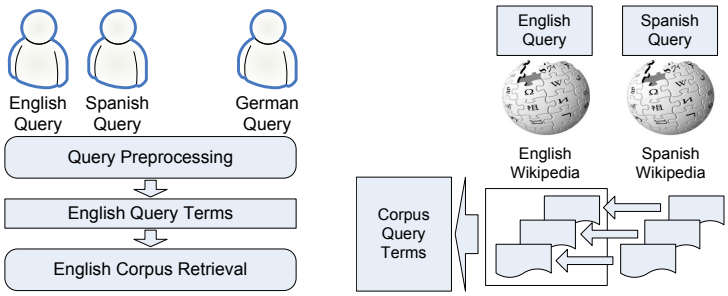
The focus in our contribution to this task, especially within this paper, lies on the following three points:

- How competitive is Apache Lucene[1], a state-of-the-art open source search engine which is used in many business applications, against scientific state-of-the-art?
- How can we reconstruct a query from a retrieval result, what does this cost for the monolingual task, and can this method be used to reconstruct the query in a different language?
- What is the impact of the provided WSD[1, 3] on this system, can we identify a statistical significant change in the performance?

In order to be able to compare different translation and disambiguation strategies we propose a fair approach to crosslanguage retrieval based on Lucene, where each query is preprocessed in the same way independent of the query and target language (Fig. 1(a)). Within our retrieval system we exploit cooccurrences

[1] http://lucene.apache.org

on corpus level to archive the cross language retrieval functionality. For our experiments in this task we used the English and Spanish Wikipedia[2]. Thereby the mapping between the articles from one language to the other language comes from the author defined cross-language links between the articles of different languages. Every query was then processed as shown in Fig 1(b): Firstly, we queried the Wikipedia index in the query language with terms from different sections of the provided queries. Secondly, we exploit the appropriate English articles from the search result and extract significant English query terms. Note that this query reconstruction step is mandatory for cross-language but is optional for monolingual problems. Thirdly, we used these query terms to query one of the CLEF indexes, either the plain index or the WSD index.



(a) Crosslanguage Retrieval (b) Cooccurrence Exploitation

Fig. 1. Retrieval Methodology

The remaining contribution is structured as follows: Section 2 provides an overview of our system in terms of index structures and methodology used. Section 3 details the proposed query processing technique. Results are outlined in Section 4 and Section 5 concludes this contribution.

2 System Architecture

Our system is based on a number of different indexes as shown in Figure 2. The Multilingual Wikipedia Index is used at the query preprocessing layer and the Plain and WSD Index are the indexes of the CLEF newspaper corpus data. The retrieval system was implemented in Java, based on the Apache Lucene text search engine library. This search engine library provides a high performance text retrieval engine for arbitrary, configurable indexes.

2.1 Multilingual Wikipedia Index

As discussed in the introduction we perform query preprocessing on every incoming query independent of the query language. The index we have build for query

[2] `http://www.wikipedia.org`

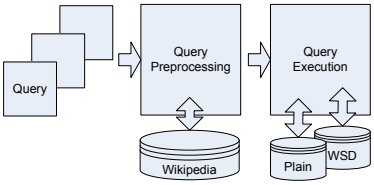


Fig. 2. System Architecture

preprocessing is called Multilingual Wikipedia Index. This index is created using English, Spanish, and possibly other Wikipedia data. Each Wikipedia article in every language is added to the Multilingual Wikipedia Index.

If one article links to another article in a different language we add an internal reference between these two articles. This approach allows to search in one language, and by exploiting the internal references, we are able to retrieve the appropriate articles in another language. To build this index the public available XML dumps are parsed with Bliki[3]. The parsed content is then indexed without further preprocessing, stemming or stop-word removal.

2.2 Corpus Retrieval Indexes

In this section we describe the different indexes we have created to retrieve the news articles from the CLEF corpus. Each of the following CLEF index contains all documents from the Los Angeles Times (1994) and Glasgow Herald (1995) dataset. Only the content of the documents was processed, title or other metadata has been ignored.

Plain Document Index For the plain text variant, the data has been processed by the default Lucene indexing chain. The newspaper plain text has been tokenized by whitespaces and then transformed to lower case. For this work we have indexed the content terms in the original word form and once in the lemmatised form, whereby we used the lemmatisation information from the dataset.

WSD Document Index For the word sense disambiguated variant, we used the available WSD information to compute the synonyms for the document terms. To maximize the impact of the WSD information we decided to only take the WordNet [6] Sense with highest WSD value from the data. All found synonym terms were indexed at the same position within the document as the original term to prevail phrase queries.

Technical speaking, the Lucene index is a document term matrix. Each document is thereby represented as a vector of terms. Lucene further allows to put more than one term on every term vector position. All terms on the same position are then transparently interchangeable. Lucene processes phrase queries

[3] `http://matheclipse.org/doc/bliki/index.html`

as follows: Firstly, all documents are searched with a boolean "and" query for all terms in the phrase. Secondly, Lucene retrieves the "distance" inbetween the terms within the term vector of all matching documents. Thirdly, if the "distance" inbetween the query terms equals one the phrase matches. Thats the reason why terms at the same position are completly interchangeable in phrase queries. For example the term *baby* and the synonym *infant* indexed on the same position makes it possible, that the phrase query *baby food* would retrieve all documents, where either *baby food* or *infant food* occurs as phrase.

3 CLEF Query Processing

First, each query is processed and interpunctation characters are removed. Next, phrase queries are identified by either underscore characters inbetween the terms or quotation marks surrounding the phrase. For phrase queries, the word order is maintained throughout the whole process. Next the query is tokenized and last stop words are removed from the query. Note that the system uses language specific stop word lists.

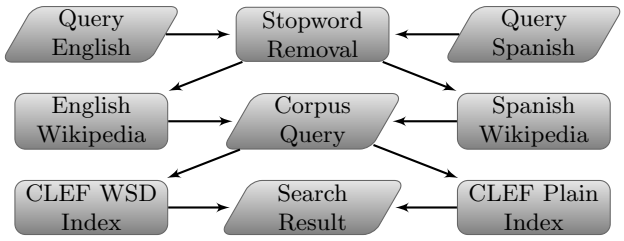


Fig. 3. Overview of the query processing

3.1 Query Translation

In the first step of query translation the extracted terms are translated to a set of terms in the search corpus target language (English). For each original query term we search in the Multilingual Wikipedia Index. We then collect the ids and the scores from the top 50 search results. Using these ids the appropriate English version of the Wikipedia articles are then retrieved by exploiting the earlier mentioned references inbetween the articles of different languages. In the next step, all English terms from these articles are extracted and we calculate a weight for each term by multiplying the score of the article with the inverse document frequency of the term. In the last step we use the top 5 terms for each separate query term to build the final query.

A major advantage of our approach is that it relies only on term distribution statistics to "translate" terms into English query terms. No additional knowledge base, like dictionaries, taxonomies, and ontoloties are used.

3.2 Query Construction

The collected and translated query terms, developed by the whole query preprocessing pipeline as shown in Fig. 3 are used to search for CLEF articles. The final query is thereby a hierarchical disjunction query. For the first level, the top 5 translated terms per original query term are used to formulate a standard boolean disjunction query. In the next level these queries are combined in two different ways: Boolean Disjunction[4] and Disjunction Max[5].

- *Boolean Disjunction:* This combination calculates the combined document score as a sum of the distinct scores for each single query term and normalizes this sum by the number of query terms. The boolean disjunction therefore calculates the mean of the scores.

$$score = \frac{1}{N} \sum s_i \tag{1}$$

- *Disjunction Max:* This combination calculates the score as the sum of the maximum score for a document for any subquery, plus a tie breaking increment for any additional matching subqueries. So the disjunction max preferes documents with high individual score. A tie breaking factor $t = 0$ leads to a total score whereby only the maximum scoring sub query contributes to the final score. In our case, where we assign importance to documents containing multiple or all query terms. That is why we have set the tie breaking factor to the number of subqueries to ensure that each combination counts more than a single retrieval result.

$$score = max_i(s_i) + t * \sum s_i - t * max_i(s_i) \tag{2}$$

Depending on the task, this hierarchical disjunction query is used to search the index with or without WSD information.

4 Experiments and Results

For this work, we have identified and solved drawbacks of our applied method for the Robust WSD Task at CLEF2008, as outlined in [4].

Motivated by the findings of other groups in the challenge and authors [5] we experimented with different retrieval features. We evaluated impact of title, description and narrative for the retrieval results. In our work we evaluated all three possible paths of the query pipeline shown in Fig. 4. From the original query we evaluated the use of different combinations of (T) title, (D) description, and (N) narrative. We further evaluated all of these pipes by using the provided lemma information instead of the wordform.

Our original system failed by a high number of queries and by looking closer to the failed queries we found a number of weird English terms in the Wikipedia

[4] org.apache.lucene.search.BooleanQuery
[5] org.apache.lucene.search.DisjunctionMaxQuery

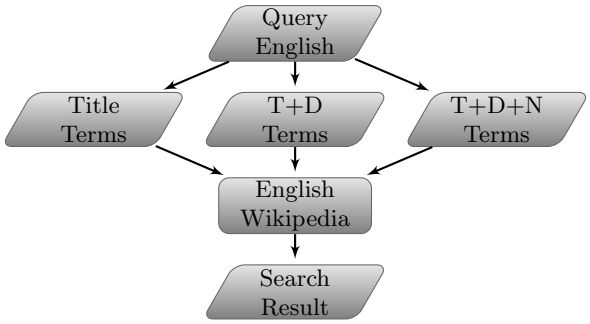
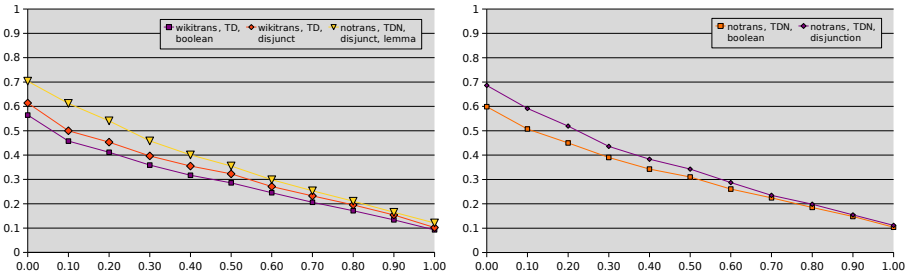


Fig. 4. Query Pipelines Wordform/Lemas

index for each article. Although such metatags should be ignored by the search engine, due to their low TFIDF[8] weight, we decided to parse the Wikipedia content to get rid of these formatting and style terms. The impact of this alteration was significant and we were able to improve our results by 3.5% for MAP(27.72 vs 32.25) as well as a consistant improvement in the precision at rank curve. This is shown in Figure 5(a); the squares curve denotes the original version and the diamond curve reflects the increased performance. The triangular curve reflects the performance of our best retrieval system including all improvements we achieved.



(a) Overall Improvement (b) Disjunction vs. Boolean Queries

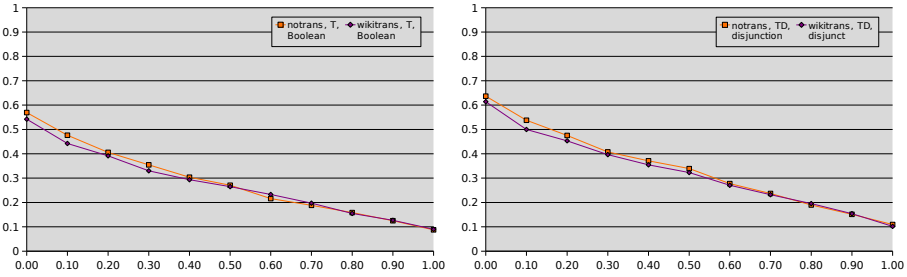
Fig. 5. Impact of Parsing and Query Combination

A drawback of the original system was that the retrieval was not fair enough for multiple matches in a hierachical boolean query. We therefore experimented with different scoring algorithms and query term combination methods. The experiments revealed that we can improve the performance by using *DisjunctionMax* queries for the monolingual task (see Fig. 5(b)) and the crosslanguage task (see Fig. 7(b)).

In combination with the use of lemmas instead of the normal wordform we were able to further improve the MAP by 1.3%. As shown in Fig. 5(a), triangle curve, we are able to outperform all earlier results with this approach. Based on our implementation we were not able to successfully apply WSD information in these experiments. Although the results are slightly better for a number of queries, we were not able to show a statistically significant improvement.

4.1 Evaluation of the Crosslanguage Methodology

The central point in our methodology is that the system is able to do fair cross language retrieval. To evaluate the cost of this approach we measured the system performance on English queries with and without Wikipedia translation. Our main intention was thereby that we are able to show that the fairness is not too expensive for the monolingual task but furthermore enables us to do cross language retrieval. As shown in Fig. 6 we were able to reveal that the query translation through Wikipedia has no significant impact on the monolingual task. As shown in Fig. 6(a) and 6(b) this holds for boolean queries as well as for disjunction max queries.



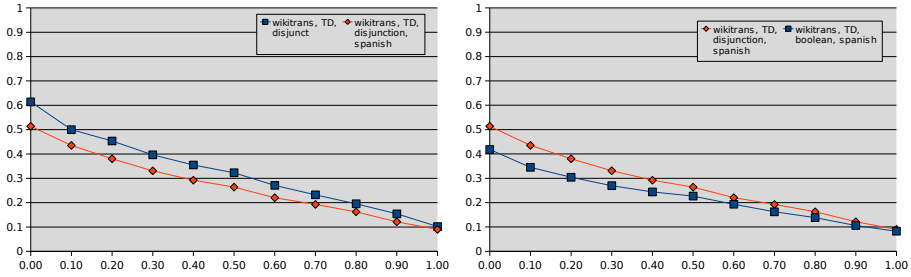
(a) Boolean Lucene Queries (b) Disjunction Lucene Queries

Fig. 6. Translation Impact

The performance of this system for the crosslingual task is shown in Fig. 7. As one can see the performance for crosslingual retrieval is worse than the performance for the monolingual task. Due to the higher degree of complexity such a result is clear. With a MAP value of about 26% we would have been competitive in the crosslangauge challange. In comparance to our best monolingual result this MAP is about 6% worse. In comparance to other groups, we are now able to work on eather task and improvements in one task will automatically improve the performance in the other task.

5 Conclusion

Our retrieval system for fair crosslanguage retrieval based on Apache Lucene has proofed to be competitive with other scientific state-of-the-art retrieval tech-



(a) Crosslanguage Performance (b) Disjunct vs. Boolean (ES)

Fig. 7. Crosslangugae Retrieval

niques with sophisitcated weighting schemes [7]. Further we were able to show that our query reconstruction methodology is not a major constraint for the monolingual task, but makes us competitive in the cross language task. In all of our experiments we were not able to show a significant improvement for the retrieval task when using word sense disambiguation information.

Acknowledgement

The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. E. Agiree and O.L. de Lacall. UBC-ALM: Combining k-NN with SVD for WSD. In *Proc. of the 4th Int. Workshop on Semantic Evaluations*, pages 341–345, 2007.
2. Eneko Agirre, Giorgio M., Di Nunzio, Nicola Ferro, Thomas Mandl, and Carol Peters. Clef 2008: Ad hoc track overview. 2008.
3. Y. Chang, H.T. Ng, and Z. Zhong. NUS-PT: Exploiting parallel texts for word sense disambiguation in the english all-words tasks. In *Proc. of the 4th Int. Workshop on Semantic Evaluations*, 2007.
4. A. Juffinger, R. Kern, and M. Granitzer. Exploiting cooccurrence on corpus and document level for fair crosslanguage retrieval. In *Working Notes for the CLEF 2008 Workshop, 17-19 September, Aarhus, Denmark*, 2008.
5. Maik Anderka: Martin Potthast, Benno Stein. A wikipedia-based multilingual retrieval model. In *Proc. of 30th European Conference on IR Research*, 2008.
6. G. Miller. Wordnet: A lexical database for english. *Comm. ACM*, 1995.
7. S. Robertson, H. Zaragoza, and M. Taylor. Simple bm25 extension to multiple weighted fields. In *Proc. of the 13th ACM international conference on Information and knowledge management*, 2004.
8. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, 1988.