Exploiting Cooccurrence on Corpus and Document Level for Fair Crosslanguage Retrieval

Andreas Juffinger, Roman Kern and Michael Granitzer
Know-Center, Graz
ajuffinger,rkern,mgranitzer@know-center.at

Abstract

In this paper we describe the methodology, architecture and implementation of the information retrieval system we have developed for the Robust WSD Task at CLEF 2008. Our system is based on an extensive query preprocessing step for homogenisation of the corpus queries. The preprocessing of queries includes: firstly, an query expansion step based on Wordnet Synonsyms or an Associative Index, secondly a query translation step based on corpus article cooccurrence in Wikipedia, and thirdly a standard disjunct index search in the CLEF corpus. The crosslanguage enabled system behaves thereby as much as possible fair over different languages. We apply the same preprocessing steps, independent of the query and corpus language, to all queries.

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; H.2.3 [**Database Managment**]: Languages

General Terms

Measurement, Performance, Experimentation

Keywords

Information Retrieval, Associative Thesaurus, Word Sense Disambiguation, Crosslanguage Retrieval

1 Introduction

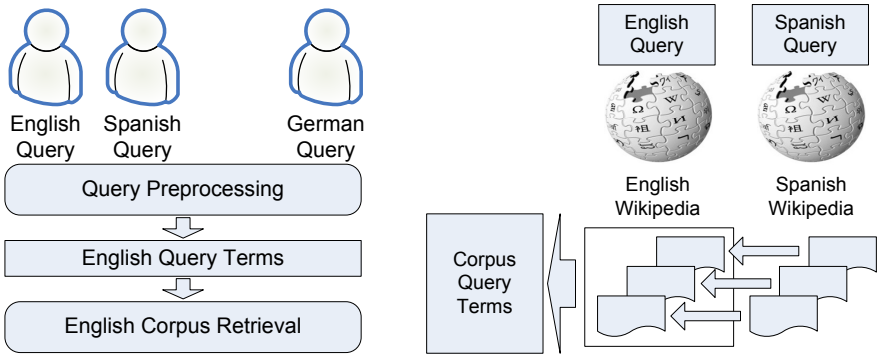
The CLEF 2008 robust task has aimed at combining semantic and information retrieval. The goal of the task was to test whether WSD can be used beneficially for retrieval systems. For this the organizers provided document collections annotated with word sense disambiguation (WSD) from previous CLEF campaigns. The english documents and the queries have been annotated by the winning WSD system from the previous year by [1] and [2] based on WordNet version 1.6. The spanish queries have been annotated by the "First Sense Heuristic" based on the spanish WordNet. The organizers believe that polysemy is among the reasons for information retrieval systems to fail.

Our approach for this task is based on the following methodology: Firstly, we build an index for the document corpus. Secondly, for each query we generate a list of hierarchical disjunct query terms in the corpus language, and thirdly, we use these query terms to search in one of the corpus index: plain document index, WSD document index, and MST network index. The plain

document index was used to calculate the result without WSD information, the WSD document index to get the task result including WSD information and at last a the MST network index for retrieval with implicetly disambiguation for baseline comparison. Altough we have deployed only a single run from our system to the challange, we discuss the idea behind the whole system architecture and lessons learned also for the crosslanguage task.

In order to compare different translation and disambiguation strategies our goal was to provide a fair approach to crosslanguage retrieval: each query is preprocessed in the same way independent of the query and target language. In other words: the system aim was to build a cross language retrieval system which is fair across different languages Fig. 1(a). The positive bias for English queries, resulting from their formulation in the corpus language, is reduced yielding to more comparable, more fair results between different languages.

Within our retrieval system we exploit cooccurrences on corpus level to achive thid fair cross language retrieval under these conditions. For the experiments in this task we used the English and Spanish Wikipedia[1]. For the mapping between the articles from one language to the other language we have extracted the cross-language links between the spanish articles and their english counterparts. Every query was then processed as shown in Fig 1(b): Firstly, we queried the Wikipedia index in the query language. Secondly, we used the appropriate english articles to extract significant english query terms. Thirdly, we used these query terms to query one of the CLEF indexes (see Sections 2.3).



(a) Fair Crosslanguage Retrieval (b) Corpus Cooccurrence Exploitation

Figure 1: Retrieval Methodology

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

2 System Architecture

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

[1] http://www.wikipedia.org
[2] http://lucene.apache.org

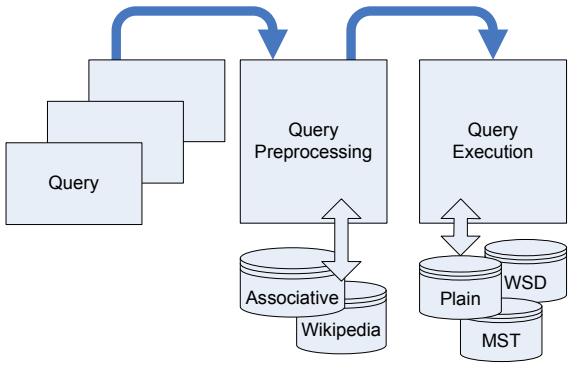


Figure 2: Sytem Architecture

2.1 Statistical Corpus Analysis

For statistical corpus analyse we have developed a dictionary modul in Java. The calculation of term weighting functions like TFIDF[4, 6] or BM25 TFIDF[5] is very data intensive and one needs many statistics from the dictionary. To calculate these weighting functions as well as to reveal other term statistics from the corpus we have tuned our dictionary to fit into memory. A simple implementation with a single document term matrix is unfeasable, due to the fact that this matrix would have a minimum size 80 bill. cells, 160000 articles multiplied by at least 500000 terms. Our implementation exploits the sparsity of the document term matrix and stores therefore only the term vectors per document. A first implementation based on `java.lang.String` revealed, that most of the memory is used to store the terms, Fig. 3(b) triangle trajectory. For calculating the statistics it is not necessary to store the terms text and therefore we implemented a disk based mapping from strings to integer values and we hold only these integer ids in memory with the effect of a tenth memory usage: Fig. 3(b) squares trajectory. With this approach we have been able to calculate all statistics in memory but with the drawback of complex reconstruction of human readable document term vectors. Due to the fact that every Java object needs at least 16bytes of memory, and strings are represented in UTF16 (4bytes per character) we decided to store the terms as UTF8 in byte arrays. The average length of the terms in this corpus is 6 characters what leads to an average term size of 6 bytes (nearly no non ascii characters occurre in english texts). Note that an integer always needs 4 bytes, so the overhead of storing the terms in byte arrays is 2 byte per term and therefore about 2 megabytes for a million terms. One can see in Fig. 3(b) diamond trajectory, that this approach is competitive with the integer id approach but with the advantage of holding all information in memory for fast document term vector reconstruction.

2.2 Query Preprocessing Indexes

As discussed in the introduction we perform query preprocessing on every incoming query independent of the query language. The indexes we have build for query preprocessing are the Multilingual Wikipedia Index and the Cooccurrence Association Index.

2.2.1 Multilingual Wikipedia Index

One important component of the system is the multilingual Wikipedia index. This index is created using the English and Spanish Wikipedia data. If an article is available in both langugae versions it is added to the multilangual Wikipedia index and an internal reference between the english and spanish text is created. Thus it is possible to search in one language and retrieve the article in the other language. To build this index the public available XML dumps are parsed and indexed. No special treatment was applied to the article text. The MediaWiki content was indexed as is without processing, stemming or stop-word removal.

2.2.2 Cooccurrence Association Index

One standard procedure applied for query expansion is the usage of term cooccurrence statistics. This approach calculates a weigthing scheme between terms based on their cooccurrence within all documents of an corpus. The basic rationale behind this calculations is the assumption that statistically significant cooccurrences have a good chance to be synonyms[3].

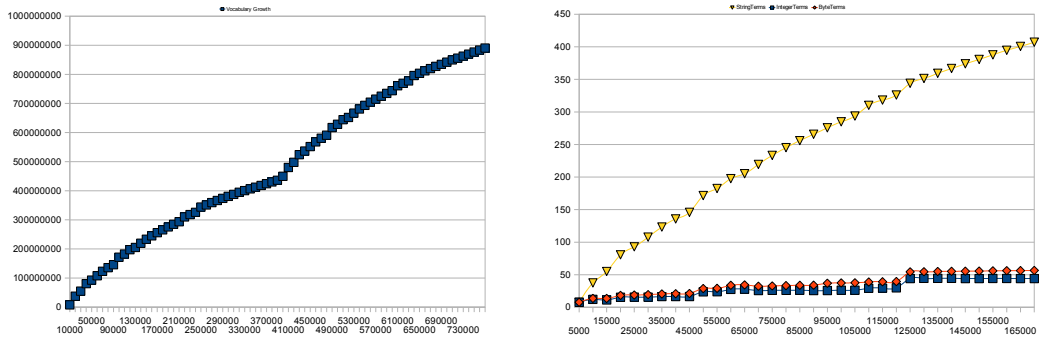
The cooccurrence network in our system is based on the Wikipedia multilingual index. For each language of this corpus (english and spanish) a separate term-term network is computed. The weights between terms are calculated by using an similarity measure based on the cosine similarity. Only cooccurrences within a predefined token vicinity are used and additionaly the distances between two terms within an Wikipedia article also contribute to the term similarity weights. Each term-term network is then stored in an associative index for fast retrieval of adjacent terms.

2.3 Robust Task 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. From the documents only the content was processed, title or other metadata has been ignored.

2.3.1 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. The dictionary containes about 600000 terms and tokens and the overall index uses 3.4G disk storage. The analysis of the term dictionary revealed a interesting artefact of the corpus in the vocabulary growth function. One easily discover that there are two different underlying datasources for the corpus (see Fig. 3(a)) - the LA Times (1994) and the Glasgow Herald (1995) with a slightly different vocabulary. A growth function for a homogen corpus would look like a single continous growing logarithmic function without an inflexion point. The reason for this inflexion point is the addition of two different logarithmic functions - one for each newspaper. The reason for the different logarithmic functions is the cultural difference and therefore a different word pool between Scotland and California.



(a) Vocabulary Growth

(b) Memory Growth

Figure 3: Corpus Growth Functions

2.3.2 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 Sense with highest WSD value from the data if the term was annotated with more than one sense. All found synonym terms were indexed at the same position within the document as the original term to prevail phrase queries. Simply speaking, the Lucene index is a document term matrix. Each document is thereby represented as a vector of terms. Lucene does further allow to put more than one term on every term vector position. All terms on the same position are then transparently interchangeable. 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. This behavior comes from the way as Lucene processes phrase queries: 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 is one, then the phrase matches. Thats the reason why terms at the same position are completly interchangeable in phrase queries. Again no additional processing other than case normalization were applied.

2.3.3 MST Network Index

For the maximum spanning tree (MST) network index we extracted, based on the part of speach (POS) tags in the corpus, only nouns from each newspaper article. Note that the focus on nouns is necessary due to the fact that otherwise the following network calculations became inpracticable and nouns carry the most valuable information. After stemming and stopword removal we ended up with an adequate list of nouns per document. To identify significant cooccurring terms α and β we test the statistical evidence by a chi-square test of independence for with the following contingency table (Tab. 1). The values A, B, C and D are the respective counts of the terms based on the scope: A the total number of terms in the corpus, B the number of occurences of the term β, C number of terms associated with the term α and D the number of terms wich occure with term α and β.

	$\forall$	β
Corpus Distribution	A	B
Cooccurring with α	C	D

Table 1: Chi-Square Contingency Table

Statistically significant tuples are then used to construct a document cooccurrence network. Each document is therefore described by a term network which contains only corpus significant term tuples where both terms occure in the document. At this level all term networks are fully connected if one takes all cooccurrences on document level into account. To prune the networks we used only the 50 top most tuples. Note that the tuple ranking is based on a corpus wide level and so the globally most significant tuples are used for each document and the fixed number of tuples leads to a corpus based "length" normalisation.

To index these networks it is necessary to serialize the term graphs into a term sequence. A general graph has no explizit start node, neither an intuitive walking strategy to visit every node. A solution to the travelling salesman problem (TSP) starting at a specific term, would lead to a reproducable term ordering, but the TSP is NP hard and therefore not computable in reasonable time. The serialization of trees is deterministic and so we transform the network in a tree and then serialize the tree in infix order. As a first heuristically transformation of the network into a tree we calculate the maximum spanning tree, based on the cooccurrence weights, of the term network. Note that higher weights stand for statistically more significant cooccurring terms. The resulting tree was then serialized in infix order starting by alphabetically lower node of the most significant term tuple.

3 CLEF Query Processing

Each query is first processed and special characters are removed, like for example interpunctation. Query terms that contain underscore charaters or terms enclosed with quotation marks are treated as phrases. For these phrases the word order is maintained throughout the whole process. After the tokenization process all query terms are removed that are found in a language specific stop word list.

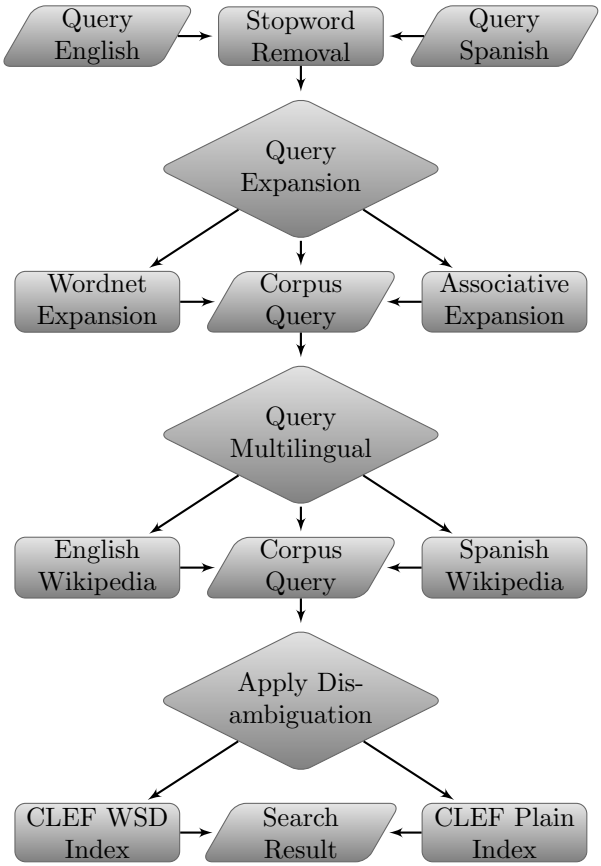


Figure 4: Overview of the query processing

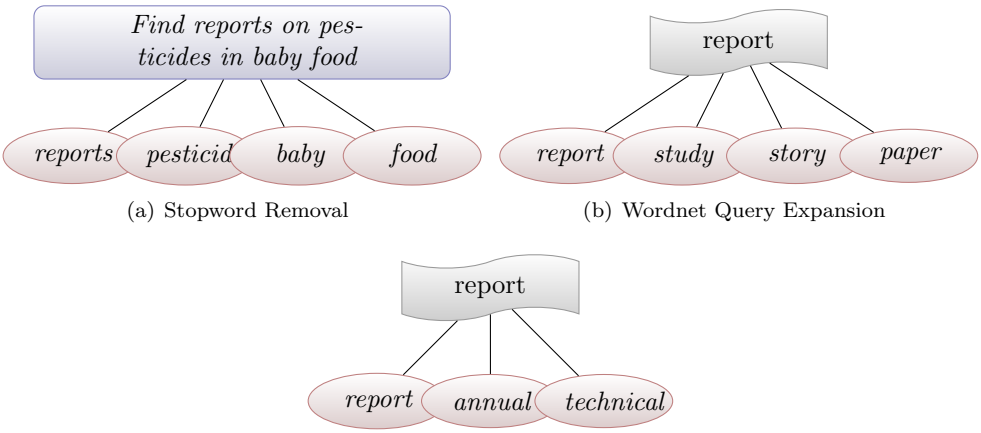
Directly after stop word removal, an optional query expansion step can take place. Depending on the configuration settings different query expansion strategies have been used. Examples for the expanded query terms are shown in Fig. 5.

3.1 Wordnet Query Expansion

The first query expansion strategy makes use of the available word sense disambiguation information. For each term that carries Wordnet synset annotations the synset with the highest score is selected. Only from this synset the synonyms form the query expansion terms. The synonyms for the spanish query terms were also taken from the english Wordnet thus no further translation processes were needed.

3.2 Associative Query Expansion

For the query expansion the individual terms are used as start nodes in the term network calculated from the cooccurrence associative index. Adjacent terms from this network were then included in



(a) Stopword Removal

(b) Wordnet Query Expansion

(c) Associative Query Expansion

Figure 5: Query Process Tree

the query, restricted by thresholds: maximum number of terms and minimal similarity value.

One can see that the associative expanded query terms are quite different for the term report. The reason for this is that the associative results are computer generated based on cooccurrence statistics and the Wordnet expansion is based on human edited synonyms. Note that we have discovered, that for other terms the results are fully "overlapping" between the association and the wordnet method. Althought extensive evaluations are depending we are confident, that the association method is very helpfull for query expansion.

3.3 Query Translation

After the query terms has been preprocessed and optionally the query has been expanded by synonyms or cooccurrences, each query terms is translated to the search corpus target language. For this task. we have applied this processing to both spanish and english queries, even if the target language is english. This is done to achieve optimal fairness for each query language.

Each term forms a search query within the multilingual Wikipedia index restricted to the language of the query. From the search result set the ids and the score of the top 50 results were collected. Using these ids the english version of the Wikipedia articles are retrieved from the index. All term from these articles are extracted and a weight for each term was aggregated. The weight for each term were calculated using the score of the article multiplied with the inverse document frequency of that term. For all terms from all top articles the aggregated weight was used to sort the result terms. From this sorted list of terms the top 5 were selected and used to build the final query for the target index.

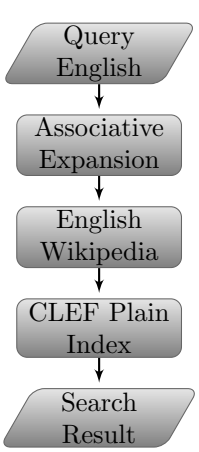
A major advantage of this approach is that it relies only on term distribution statistics to"translate" english and spanish terms into english query terms. No additional knowledge base, like an dictionary are used. Note that the presented results have been reached without extensive tuning.

3.4 Search CLEF Articles

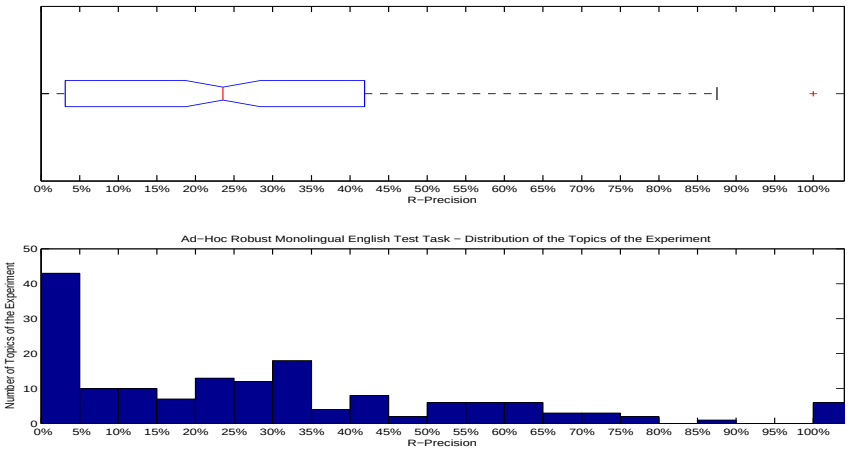
The collected and translated query terms, developed by the whole query preprocessing pipeline as shown in Fig. 4 with or without the optional query expansion and multilingual steps, are then used to build up a hierarchical disjunction query. Depending on the task either the index with or withoud WSD information was then searched with this disjunction query. The default Lucene scoring algorithm, based on TFIDF, was used to calculate the rank of the retrieved articles.

4 Experiments and Results

As one of our best system configuration we have applied at the CLEF 2008 Robust Task the Associative Pipeline as shown in Fig. 6(a). For this pipeline we used the associative query expansion modul, the english Wikipedia translation modul. The query terms developed by this process haven then been applied as disjunct query terms to the CLEF Plain Index with the results shown in Fig. 6(b). As mandatory by the task we also have applied the WSD Version of this run with the pipeline shown in Fig. 7(a) and the results shown in Fig. 7.



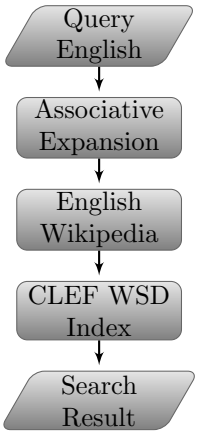
(a) Query Pipeline Associative Retrieval



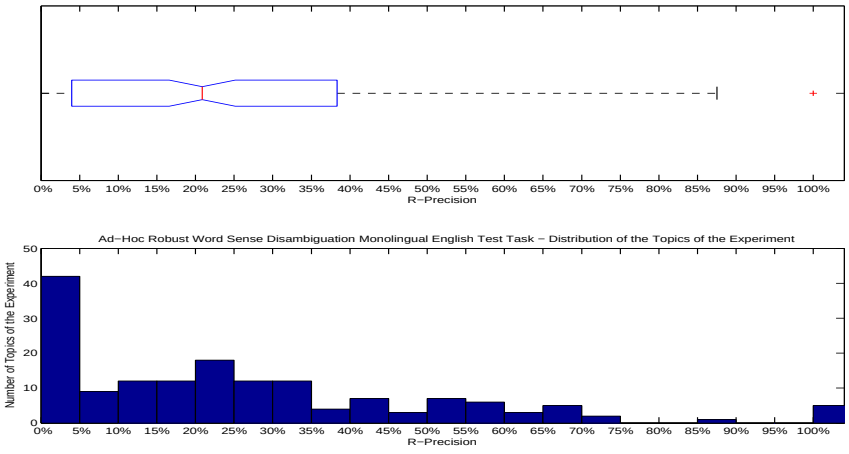
(b) Comparison Median R-Precision

Figure 6: Associative Retrieval without WSD

Based on the original system implementation we haven't been able to successfully apply WSD information. Although the results are slightly better for a number of queries, we have not been able to show a statistically significant improvement.



(a) Query Pipeline Associative Retrieval

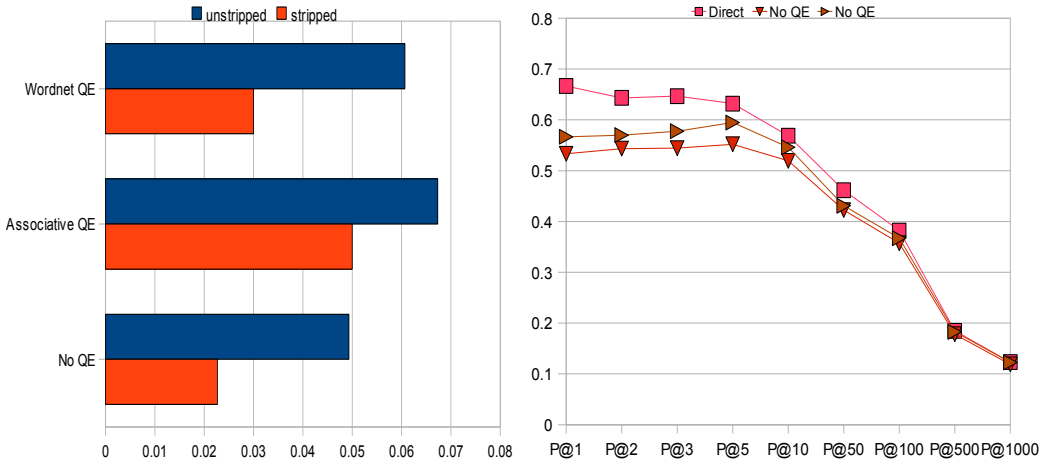


(b) Comparison Median R-Precision

Figure 7: Associative Retrieval with WSD

Motivated by the fact that our system failes by a hight number of queries we tried to figure out whats the reason for this. We have identified a bug in our system in the preprocessing pipeline of the Wikipedia Index, which we have fixed for later runs. Deflated by the suboptimal results

we also evaluated the impact of the methodology of beeing fair to every language. As shown in Fig. 8(a) started to evaluate the impact of the earlier noted bug in our implementation. Without the improvements we lost about 6 to 7 percentage precision for our Wikipedia methodology in comparance to "direct" retrieval without the Wikipedia normalisation. Unfortunately, the impact of the parsing problem biased our applied result significantly. The improved system is now about 2 to 3 percentage points worse than the "direct" retrieval baseline, and therefore we know that our system can perform significantly better. One can see in Fig. 8(b) our system now performs always better than the earlier version, what makes us confident that will be able to resolve the remaining difference, or in other words the costs for the methodology will be insignificant.



(a) Negligible Methodology Impact

(b) Precision at Rank Comparison

Figure 8: Improved Results

4.1 Evaluation of the Crosslanguage Methodology

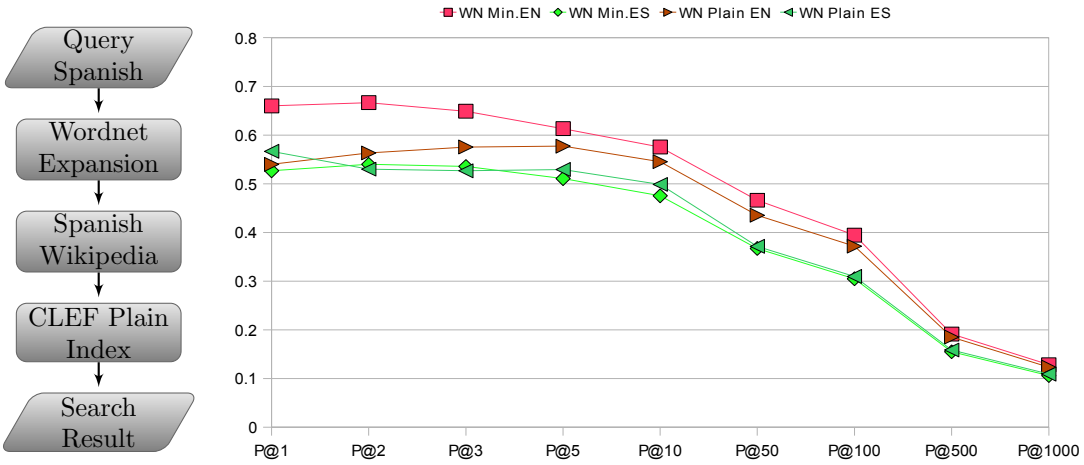
The central point in our methodology is that the system is able to do fair cross language retrieval. Therefore we have evaluated the system performance on spanish queries. For this test we have used the pipeline shown in Fig. 9(a) with Wordnet Query Expansion and the Spanish Wikipedia and Plain CLEF Index. The performance of the retrieval task is shown in Fig. 9(b). Again the "optimal" performance of the system is the "direct" english method *WN Min.EN*.

The crosslanguage retrieval performance is shown in this Figure as *WN Plain.ES*. One can see that this crosslingual system performance ist nearly the same as the *WN Plain.EN* monolingual performance, from which one can derive, that our system is, as intended, also competitive for the crosslingual case.

5 Conclusion

In this working note we have outlined our retrieval system for fair crosslanguage retrieval. Although we have applied only on the monolingual task we have been able to show, that our system perform as good as for the monolingual task in the crosslingual setup. Further we have shown that this fairness condition is not a major constraint. For the applied results we have not been able to show a significant improvement for the retrieval task when using word sense disambiguation information. The reason for this is probably, that the system is still not optimal for the retrieval task without WSD so the expected improvement through WSD might be lost by still undiscovered systematical faults of the system.

One of our next steps will be a deeper analysis of a number of Wikipedia articles as cooccurrence corpus. Although we have already looked up a number of articles we have not been able to develop



(a) Query Pipeline Crosslingual Retrieval

(b) Crosslanguage Retrieval Comparison

Figure 9: Crosslingual Retrieval Performance

explezit characteristics to select the optimal cooccurring articles from the Wikipedia. Another important improvement to our system would be a more sophisitcated preprocessing module for our indexes.

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] 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.

[3] R. Kern, M. Granitzer, and V. Pammer. Extending folksonomies for image tagging. In *WIAMIS 2008 , Special Session on Multimedia Metadata Management andRetrieval, IEEE Computer Society, Klagenfurt*, 2008.

[4] C. J. Rijsbergen. *Information Retrieval, 2nd Edition*. Butterworths, 1979.

[5] S.E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *In Proc. of the ACM SIGIR conference on research and development in information retrieval*, 1994.

[6] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, 1988.