ImageCLEF 2006 Experiments at the Chemnitz Technical University

Thomas Wilhelm, Maximilian Eibl

Chemnitz University of Technology
Faculty of Computer Science, Media Informatics
Strasse der Nationen 62, 09107 Chemnitz, Germany
thomas.wilhelm@s2000.tu-chemnitz.de, eibl@informatik.tu-chemnitz.de

Abstract. We present a report about our participation in the ImageCLEF photo task 2006 and a short description of our new framework for future use in further CLEF participations. We described and analysed our participation in the monolingual English task. Special Lucene-aligned query expansion and histogram comparisons are helping to improve the baseline results.

ACM Categories and Subject Descriptors

H.3.1 - Content Analysis and Indexing, H.3.3 - Information Search and Retrieval, H.3.4 - Systems and Software

Keywords

image retrieval, cross-language information retrieval, query expansion

1 Introduction

Our goal was to develop a new information retrieval framework which can handle different custom retrieval mechanisms (i.e. Lucene, the GNU Image-Finding Tool and others). The framework should be so general that one can use either one of the custom retrieval mechanisms or combine different ones. Therefore our efforts were not focused on better results but to get the framework up and running.

2 System description

For our participation at CLEF 2006 we constructed a GUI (graphical user interface) quite easy to handle. It bases on Lucene version 1.4 and includes several user-configurable components. For example we wrote an *Analyzer* in which *TokenFilter*s can be added and rearranged.

For the participation in ImageCLEF 2006 we needed to extend this system with image retrieval capabilities. But instead of extending the old system we rewrote it and developed a general IR framework. This framework does not necessarily require Lucene and makes it possible to build an abstract layer for other search engines like GIFT (GNU Image-Finding Tool) and others.

The main component of our framework is the class *Run*, which includes all information required to perform and repeat searches. A *Run* contains

- the *Topic*s, which can be preprocessed by *TopicFilters*
- an *Index*, which was created by an *Indexer* from a *DataCollection*
- a search method called *Searcher*, which can read and search the *Index*
- and *HitSet*s, which contain the results.

To use another search engine one will have to extend the abstract classes *Indexer* and *Searcher*. To use another data collection type (e.g. GIRT4, IAPR, ...) one will have to extend the abstract class *DataCollection*.

In order to merge results there is a special extension of the class *Searcher*: the *MergeSearcher*. It combines several *Searcher*s and uses an abstract class *Merger* to merge their results.

2.1 Lucene

The text search engine used is Lucene version 1.9 with an adapted *Analyzer*. As described above we create a *LuceneIndexer* and an abstract *LuceneSearcher* that provides basic abilities to underlying *Searcher*s like a selectable Lucene *Analyzer*. The *Analyzer* is based on the *StandardTokenizer* and *LowerCaseFilter* by Lucene and is enhanced by a custom Snowball filter for stemming. We also added a positional stop word filter and used the stop word list suggested on the CLEF website. The *Analyzer* details (stemmer and stop word list) can be configured through a GUI. The *LuceneStandardSearcher* works with the Lucene *MultiFieldQueryParser* and searches in specified fields of the index.

2.2 GIFT (GNU Image-Finding Tool)

The GIFT was not used for a submitted run but it should be mentioned here to show the abilities of the framework. We only implemented a *Searcher*, which can access a GIFT server and takes images to generate queries by example. The underlying index must be created with the command line utility provided by the GIFT.

2.3 IAPR Data Collection

To access the IAPR data collection of ImageCLEF we implemented the class *IaprDataCollection* that provides two special options for the *DataDocument*s to contain special data:

- option 1: the *DataDocument*s contain the annotations
- option 2: the *DataDocument*s contain the histograms

In both cases the associated images can be retrieved with the help of the *DataDocument*.

3 Description of the runs submitted

We focused on the monolingual task using the English topics and annotations.

The Lucene index of the annotations contains all fields. All fields are indexed and tokenized except DATE, which is not tokenized. The DOCNO is the only field which is stored, resulting in a quite small index (only 2 MB).

We used the default *BooleanQuery* from Lucene with no further weighting of any field pairs in order to avoid any corpus specific weighting.

The default Lucene search classes (default similarity, etc.) were used to search the topic-annotation field pairs as shown in table 1.

Topic field	*Annotations searched*
TITLE	TITLE, LOCATION, DESCRIPTION, NOTES
NARR	DESCRIPTION, NOTES

Table 1. Topic-annotation field pairs

Our mechanism of query expansion is adapted to the Lucene-index and is, though simple, a little bit slow. First of all a list of assumable relevant documents is identified. Then the Lucene-generated index is checked for terms also contained in documents in this list. These terms, their frequency and the number of documents in which they are found are stored in a second list. Frequency is used to apply a threshold: only terms with a high frequency are used to expand the query. The new terms are weighted by the ratio of its frequency and the number of containing documents.

For all runs, except “tucEEANT”, the query expansion was applied once, using the first 20 results of the original query.

In the run “tucEEAFT2” the query expansion is applied twice. The first time using the annotations of the example images and the second time using the first 20 results of the original query.

To speed up the colour histogram comparison, we also created an index for the histogram data. For each image

the histograms for the three components hue, saturation and brightness are stored and can be retrieved by the DOCNO. Though we use a Lucene index, it is used more like a database.

For the run "tucEEAFTI" all results of the textual search are scored using the Euclidean distance of their colour histograms. The histogram distance makes 30% of the new score. The remaining 70% are taken from the original score.

4 Results

Table 2 shows our results at ImageCLEF 2006. To compare our results, the top four runs of the monolingual English task are included in table 2.

Rank EN-EN	*Rank ALL*	*System*	*MOD*	*A/M*	*FB*	*QE*	*MAP*
1	1	Cindi_Exp_RF	MIXED	MANUAL	WITH	YES	0,3850
2	2	Cindi_TXT_EXP	TEXT	MANUAL	WITH	YES	0,3749
3	3	IPAL-PW-PFB3	MIXED	AUTO	WITHOUT		0,3337
4	5	NTU-EN-EN-AUTO-FB-TXTIMG	MIXED	AUTO	WITH		0,2950
5	16	**tucEEAFT2**	TEXT	AUTO	WITH	YES	0,2436
17	33	**tucEEAFTI**	MIXED	AUTO	WITH	YES	0,1856
18	34	**tucEEAFT**	TEXT	AUTO	WITH	YES	0,1856
23	47	**tucEEANT**	TEXT	AUTO	WITHOUT		0,1714

Table 2. Ranks of our submitted runs and the first four positions

As one can see, run "tucEEAFT2" is our best run which is no surprise if one takes into account that three definitely relevant documents are used for query expansion. Our second best run is "tucEEAFTI" which shows that the additional histogram comparison produces better results than without it.

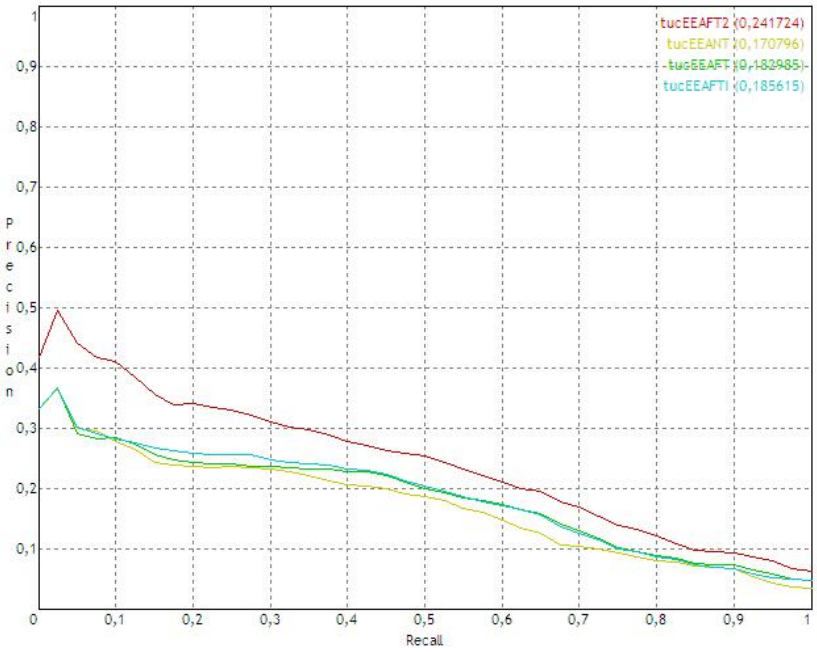

Fig. 1. Recall-precision graphs for our submitted runs

5 Conclusion

The good results in ImageCLEF 2006 show that our framework works and that our first implemented search methods provide good results.

The normal query expansion improves the mean average precision by more than 1%. Considering the poor overall mean average precision this seems to be a good improvement of the results. Although the histogram comparison is a very simple and presumably poor measurement to compare images, it improves our text-only retrieval slightly.

The next steps to improve the results is to implement a better image retrieval algorithm due to the weak performance of the histogram comparison.

References

1. Stevens, J. S., Husted T., Cutting D., & Carlson P. (2005). *Apache Lucene - Overview - Apache Lucene*. Retrieved August 10, 2006, from http://lucene.apache.org/java/.
2. Grubinger M., Leung C., & Clough P. (2005). *The IAPR Benchmark for Assessing Image Retrieval Performance in Cross Language Evaluation Tasks*. Retrieved August 10, 2006, from http://ir.shef.ac.uk/cloughie/papers/muscle-imageclef2005.pdf.
3. Roos, M. (2005). *The GNU Image-Finding Tool - GNU Project - Free Software Foundation (FSF)*. Retrieved August 10, 2006, from http://www.gnu.org/software/gift/.