Abstract
This article describes the Xtrieval framework - a flexible platform for testing and evaluating different aspects of retrieval systems. The background for the design of the framework is to develop an extensible platform for the combination of text and content-based multimedia retrieval. We describe the general composition of the framework and its main functionality. An evaluation of the framework capabilities is given with two exemplary retrieval evaluation tasks.

1 Introduction
In 1992 the first evaluation conference on methods for text retrieval TREC [Harman, 1992] was held. About the turn of the millennium a few groups separated from TREC because of more specific evaluation purposes. They can mainly be divided into multilingual text retrieval campaigns like CLEF [Peters, 2000] or NTCIR [Kando, 1999] and into the multimedia retrieval community with conferences like TRECVID [Smeaton et al., 2001] or ImageCLEFphoto [Grubinger et al., 2006].

Three years ago, the Chemnitz retrieval group implemented a first prototype for information retrieval evaluation [Kürtston and Eibl, 2006; Wilhelm and Eibl, 2006] for the participation in the CLEF campaign. This prototype provided functionality to prepare and execute experiments corresponding to the given retrieval tasks. The analysis of the results motivated us to develop a general framework for experimental retrieval evaluation. The main goals for the implementation were flexibility and extensibility.

Nowadays, there is a number of toolkits for experimental retrieval evaluation available, especially in the text retrieval domain (e.g. Terrier [Ounis et al., 2007], the Lemur toolkit1 or the Zettair search engine2). These platforms provide research tools for evaluating new ideas in the text retrieval domain with respect to commonly used state-of-the-art approaches. Otherwise, a very useful content-based image retrieval (CBIR) systems exist, like FIRE [Deselaers et al., 2005], GIPT3, LIRE4 and CIRES [Iqpal et al., 2002]. But to the authors knowledge there are no comparable systems available today, that combine CBIR and classical textual IR.

2 Motivation
The Xtrieval framework is part of the project sachsMedia - Cooperative Producing, Storage and Broadcasting for Local Television Stations at Chemnitz University of Technology. This project does research in two fields: automatic annotation and retrieval of audiovisual media on the one hand and distribution of audiovisual media via digital video broadcasting (DVB) and IPTV on the other hand.

The annotation of the material is carried out intellectually according to principles of formal documentation. Alternatively, sophisticated methods of multimedia retrieval, like object recognition and automated speaker recognition, will be implemented in our content-based analysis and annotation framework.

In order to enable local TV stations to cooperate with each other, a common database is set up. Within this database raw, produced and broadcast-ready material is stored by every associated TV station. This material needs to be described as comprehensively as possible in order to be easily accessible. We will use the Xtrieval framework to provide this accessibility by managing, accessing and retrieving the audiovisual content via metadata descriptions.

2.1 Content-based Audiovisual Analysis Framework
We use the Java Media Utility (JMU) framework5 based on the FFMPEG library6 to extract the frames and to implement methods for the extraction of text. JMU was developed to provide an easy way of integrating new algorithms for image or frame processing.

The framework itself consists of three main components:
- C++ Media Utility (CMU) to access the methods from FFMPEG library,
- JMU-JNI bridge to access the CMU code in Java,
- JMU framework code itself.

The main advantage of the framework is its flexibility, e.g. one can easily represent a text information extraction architecture as presented in [Jung et al., 2004] by constructing a specific processing chain. The linked components could be represented no matter if their structural connection is flat or hierarchical. Another important feature is the multiple instance architecture that allows the framework to feed several process components with the same data source in a way that parallel computation could be realized. We will use the framework not only for the use case of text recognition, but also for more sophisticated image processing applications as well as for audio or speech processing.

1http://www.lemurproject.org
2http://www.seg.rmit.edu.au/zettair
3http://www.gnu.org/software/gift
4http://www.semanticmetadata.net/lire
5http://jmu.sourceforge.net
6http://ffmpeg.mplayerhq.hu
2.2 Xtrieval Framework

The basic idea of the Xtrieval framework is to provide interfaces to combine different state-of-the-art text retrieval techniques on the one hand and to evaluate and integrate new methods for multimedia retrieval on the other hand. A complete description of the framework design process and some first evaluations is given in [Wilhelm, 2008].

For the implementation of the Xtrieval framework we used our experiences with former prototypes and specified the following requirements:

- **Object-orientated API Specification**
  The general API should provide interfaces to methods that are necessary for all possible designs of retrieval systems. Thereby we can guarantee the ability to exchange, evaluate and combine different components of retrieval systems.

- **Usage of Other Retrieval Toolkits**
  The new system must not be constrained to only one retrieval system and hence allow the use of different retrieval systems. Beyond that, the framework supports the combination of results of different retrieval systems.

- **Evaluation Component**
  In line with the prototype implementation, a component for experimental retrieval evaluation is needed. This incorporates methods for the calculation and visualization of recall-precision graphs. Additional functions to load and save relevance assessments in popular formats (e.g. TREC) are provided. A supplementary interface should be designed to enter and manage additional relevance assessments.

- **Simple but Flexible Configuration**
  We intend to create a general configuration interface that supports the user in configuring all parameter driven classes in the graphical user interface. Thus all parameters of each class can be changed during runtime without any changes in the source code of the project.

3 Framework Description

The Xtrieval framework has four main components: Indexing, Retrieval and Evaluation (the system core) on the one hand and the graphical user interface on the other hand. The system core provides elementary methods for the retrieval process and defines the interfaces for all additional functions. The graphical user interface allows the user to access the functions of the framework and visualizes diagrams from the characteristic numbers that were calculated by the framework. A schematic representation of the Xtrieval framework is illustrated in figure 1.

3.1 Indexing

A general programming interface is needed to convert a data collection into a database structure that could be used by a retrieval system. Therefore, we defined a layer that represents the real data. We implemented a simple programming interface that allows to transform every structured data collection into that representation.

The next step during indexing is the token analysis and transformation. Again, a flexible programming interface is used to access all commonly used transformation and tokenization procedures like snowball stemming algorithms or simple stop-word removal.

Finally, we created a layer to write to an index of a retrieval toolkit. This programming interface allows us to integrated famous retrieval toolkits of the IR community, as already was mentioned in the introduction. At the moment Lucene\(^7\) for textual retrieval and PostgreSQL\(^8\) for CBIR are used as retrieval cores, because they combine effectiveness and efficiency and is widely used in real-world applications.

3.2 Retrieval

We also developed a general programming interface for the retrieval component. At first there is a query preprocessing component, which could be used for the token analysis, but also for translation or any kinds of query transformation.

The main contribution of the Xtrieval framework is the flexible searcher programming interface. It allows us to use retrieval algorithms of common frameworks. But it is also possible to combine search in different indexes and combine these results into one result set. For the combination we created another programming interface, where one can simply implement an algorithm for result list fusion. Three standard approaches are implemented so far: (a) Sum-RSV, (b) Product-RSV and (c) Z-Score. Again, just like the index creation, we use Lucene as retrieval core. Because of the design of the retrieval programming interface it is even possible to combine retrieval results from different retrieval platforms, as soon as their specific query languages were implemented.

Another important feature of the retrieval component is its connection to the evaluation component and the graphical user interface. It allows us to test and evaluate new retrieval approaches on the fly. The connection to the graphical user interface allows us to run interactive retrieval experiments, because we can use feedback from the user.

3.3 Evaluation

The evaluation component includes several elements for the visualization of retrieval effectiveness like recall-precision graph or box-plot.

The most essential functionality of the evaluation component is its capability to store and reload experiments and their complete parameter sets. This enables us to repeat experiments at a later date. Additionally, it is possible to load and store relevance assessments in the TREC format.

\(^7\)http://lucene.apache.org
\(^8\)http://www.postgresql.org
4 Experimental IR Evaluation

Since the Xtrieval framework was designed for experimental information retrieval research the evaluation of retrieval approaches remains its main use case. With the obtained flexibility other use cases like the implementation of a search engine application are possible. Two exemplary implementations of special evaluation tasks based on the Xtrieval framework are described in the following subsections. Both achieved remarkable results in the corresponding evaluation task of the CLEF campaign. Another interesting application is the management tool for relevance assessments, which is described in the last subsection.

4.1 Pseudo-Relevance Feedback based on Classification

The main idea of this work was to evaluate whether term or document clustering could improve retrieval effectiveness, when it is utilized in a pseudo-relevance feedback (PRF) algorithm. Therefore, we implemented several clustering approaches and adapted them to our PRF mechanism. The whole project is described in [Kürsten and Eibl, 2006] and [Kürsten, 2006].

Local Clustering

As already mentioned, we applied clustering in our PRF procedure. In our standard PRF algorithm the feedback terms were extracted from the top-10 documents of an initial retrieval run. Several clustering algorithms were compared [Kürsten, 2006] for the application in the PRF procedure. The modified PRF with classification (PRF-C) uses the top-10 documents from the standard PRF and the documents marked as relevant by the clustering algorithm. The maximum number of documents for PRF-C is 20, but it depends on the actual overlap between the top-10 standard PRF documents and the top-10 PRF-C documents, i.e. the actual number is dynamic and varies between 10 and 20.

Experimental Setup

For our experiments we used the following configurations for PRF: (a) document clustering (LDC): using the top k documents returned from clustering the top n documents for PRF, (b) term clustering (LTC): using the terms returned from clustering the terms of the top n documents for PRF, (c) PRF-CD: combining LDC with PRF and (d) PRF-CT: combining LTC with PRF.

For LDC some preliminary experiments showed, that clustering the top n = 50 documents and returning k = 10 probably relevant documents of an initial retrieval provides the best results. In the LDC clustering procedure the k-means algorithm was implemented for all experiments shown in this evaluation. The LTC approach is a little different from LDC, because the objects to be clustered are not the top k documents, but the terms of those top k documents. To get reliable term correlations, we used the top k = 20 documents for LTC.

Experimental Results

Table 1 lists the results of clustering experiments we conducted for the participation at the monolingual subtask of the CLEF 2006 domain-specific track [Stempfhuber and Bärisch, 2006]. PRF with classification is compared to our baseline results (1st row) and a standard PRF approach (2nd row). Finally, we compare the results of two further experiments. Both combine different stemming approaches for German, but for the first run (PRF*) standard PRF was used while for the second experiment (PRF-CD*) the PRF-CD approach was applied as described before.

The results show that PRF-CD can achieve a small improvement in mean average precision (MAP) and mean recall (MR). Furthermore, it is obvious that LTC or PRF-CT did not improve precision or recall.

4.2 Multi-modal Image Retrieval

For the participation in the ImageCLEF photographic retrieval task we had to combine textual retrieval with CBIR. The corpus for the ImageCLEF photographic retrieval task was the IAPR TC-12 benchmark [Grubinger et al., 2006], which consists of 20,000 annotated photographs in a touristic context. Annotations were available in English, German and Spanish language. There were 60 topics in 16 different languages with 3 example photos each.

Experimental Setup

Lucene, our standard text retrieval component, was combined with a CBIR approach using common MPEG-7 descriptors. Each of the following descriptors was calculated using Caliph & Emir and stored in an extra database: scalable color, dominant color, color layout and edge histogram descriptor. In the experiments two modalities were tested: text only (text) and the combination of text and images (mixed).

Additionally, a thesaurus based query expansion was used, which did not improve the results in most cases. But in combination with a manual relevance feedback it was always able to produce better results. The feedback was applied to the first 20 results.

Experimental Results

Table 2 shows the results of selected experiments. Both modalities compared show that the combination of text retrieval and CBIR results in a better performance.

Table 2: Comparison of text retrieval performance

<table>
<thead>
<tr>
<th>identifier</th>
<th>MAP</th>
<th>MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>0.3811</td>
<td>0.6674</td>
</tr>
<tr>
<td>PRF</td>
<td>0.4603</td>
<td>0.8154</td>
</tr>
<tr>
<td>LTC</td>
<td>0.4408 (-4.36%)</td>
<td>0.8024 (-1.59%)</td>
</tr>
<tr>
<td>PRF-CT</td>
<td>0.4394 (-4.54%)</td>
<td>0.7968 (-2.28%)</td>
</tr>
<tr>
<td>LDC</td>
<td>0.4603 (+0.0%)</td>
<td>0.7949 (-2.51%)</td>
</tr>
<tr>
<td>PRF-CD</td>
<td>0.4726 (+2.67%)</td>
<td>0.8225 (+0.87%)</td>
</tr>
<tr>
<td>PRF*</td>
<td>0.5454</td>
<td>0.8534</td>
</tr>
<tr>
<td>PRF-CD*</td>
<td>0.5632 (+3.26%)</td>
<td>0.8912 (+4.43%)</td>
</tr>
</tbody>
</table>

Table 2: Comparison of image retrieval performance

<table>
<thead>
<tr>
<th>run id</th>
<th>lang</th>
<th>mod</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-txt</td>
<td>en</td>
<td>text</td>
<td>0.1357</td>
</tr>
<tr>
<td>en-mix</td>
<td>en</td>
<td>mixed</td>
<td>0.1682 (+24%)</td>
</tr>
<tr>
<td>en-mix-mix-qe-fb20</td>
<td>en</td>
<td>mixed</td>
<td>0.2908 (+114%)</td>
</tr>
<tr>
<td>es-txt</td>
<td>es</td>
<td>text</td>
<td>0.1208</td>
</tr>
<tr>
<td>es-mix</td>
<td>es</td>
<td>mixed</td>
<td>0.1601 (+33%)</td>
</tr>
<tr>
<td>es-mix-mix-qe-fb20</td>
<td>es</td>
<td>mixed</td>
<td>0.3069 (+153%)</td>
</tr>
<tr>
<td>en-es-txt</td>
<td>en-es</td>
<td>text</td>
<td>0.1199</td>
</tr>
<tr>
<td>en-es-mix</td>
<td>en-es</td>
<td>mixed</td>
<td>0.1582 (+32%)</td>
</tr>
<tr>
<td>en-es-mix-mix-qe-fb20</td>
<td>en-es</td>
<td>mixed</td>
<td>0.3057 (+155%)</td>
</tr>
</tbody>
</table>

9http://sourceforge.net/projects/caliph-emir
It also can be observed that the performance improves even more if the text only result has a low mean average precision.

The best results were achieved with a combination of all evaluated improvements: query expansion, manual relevance feedback and the combination of text and image retrieval (see Table 2, run id: es-mix-qe-fb20).

4.3 Relevance Assessment

Relevance assessments for retrieval experiments can easily be displayed and imported or exported to the TREC format. A programming interface to provide access to the real data of the corpus is integrated. In the retrieval evaluation use case this property seems to be very useful, because one can simply generate additional relevance assessments. Additionally, the framework can gather user relevance feedback when it is used as multimedia search engine, since user relevance feedback is one of the most successful approaches to close the semantic gap between media content and human media descriptors.

5 Future Work

For future enhancements we will concentrate on the two main use cases, which are experimental retrieval evaluation and the multimedia search engine application. For the continuation in the field of experimental retrieval evaluation we plan or actually work on the following aspects:

- integration of the Lemur and Terrier [Ounis et al., 2007] toolkits,
- analysis and evaluation of automatic feedback approaches,
- creation of a parameter optimization component,
- design and implementation of a question answering component,
- implementation of a web-based search engine application.

Another important step will be the design and implementation of a programming interface between the XTRIEVAL framework and our annotation and analysis framework.

Acknowledgments

This work was partially accomplished in conjunction with the project sachsMedia, which is funded by the Entrepreneurial Regions program of the German Federal Ministry of Education and Research.

References


