XML Information Retrieval from Spoken Word Archives

Robin Aly, Djoerd Hiemstra, Roeland Ordelman, Laurens van der Werff, and Franciska de Jong

University of Twente, The Netherlands
{r.aly, d.hiemstra, r.j.f.ordelman, l.b.vanderwerff, f.m.g.dejong}@utwente.nl

Abstract. In this paper the XML Information Retrieval System PF/Tijah is applied to retrieval tasks on large spoken document collections. The used example setting is the English CLEF-2006 CL-SR collection together with given English topics and self produced Dutch topics. The main findings presented in this paper are the easy way of adapting queries to use different kinds and combinations of metadata. Furthermore simple ways of combining different metadata kinds are shown to be beneficial in terms of mean average precision.

1 Introduction

Traditionally the retrieval of information from a multimedia document collection has been approached by creating indexes relying on manual annotation. With the emergence of digital archiving this approach is still widely in use and for many archiving institutes the creation of manually generated metadata is and will be an important part of the daily work. When the automation of metadata generation is considered, it is often seen as something that can enhance the existing process rather than replace it. The available metadata will therefore often be a combination of highly reliable and conceptually rich manual annotations, and (semi-) automatically generated metadata. A big challenge for Information Retrieval is to combine the various types of metadata and to let the user benefit from this combination.

Audio metadata may be regarded as a set of special document features; each feature adds to the overall representation of a document. Traditionally, these features include for example locating the speech/non-speech fragments, the identification of the speaker and production of full text transcripts of the speech within the document. In a lot of cases, speech recognition transcripts even provide the only means for searching a collection, for example when manual annotation of a collection is not feasible. More recently tools have become available for the extraction of additional speaker features from the audio signal such as emotions (e.g., affect bursts such as sobs, crying or words that express emotions), language, accent, dialect, age and gender.

Unlike with text content the notion of a document is not so obviously defined. A document, comprising for example a whole interview, can be very long

C. Peters et al. (Eds.): CLEF 2006, LNCS 4730, pp. 770–777, 2007.

and probably not every part is relevant given a specific user request. Browsing manually through a long multimedia document to find the most relevant parts can be cumbersome. Therefore, multimedia retrieval systems have to be capable of retrieving only relevant *parts* of documents. As a consequence partitioning or segmentation of documents should take place prior to the actual extraction of features.

Usually, the segmentation algorithm for defining sensible document parts comes almost naturally with a document collection. In the English part of the Multilingual Access to Large Spoken Archives (MALACH) collection, which is the target collection for CLEF's CL-SR task, parts of interviews on a single coherent subject have been selected as a segments. In a broadcast news retrieval system, single news items would be the obvious unit to choose for segmentation purposes.

However, given that the currently available techniques can generate multiple annotations, multiple segmentations should be supported. This complicates the task. For the MALACH collection an Information Retrieval system should be able to handle the following type of query: "give me document parts where male, native Dutch speakers are talking about [...] without expressing any emotions". Thus the retrieval system needs to consider various sources to yield a good performance. With this scenario the following questions need to be addressed: (i) how to efficiently store these multiple annotations, (ii) how to combine and rank retrieval scores based on them and (iii) how to select appropriate parts of the documents that are going to be returned as part of a result.

MPEG-7 [1] defines standardized ways of storing annotations of multimedia content. It has the potential of solving some of the issues related to the use of multiple annotations in multimedia retrieval. For this reason the underlying XML [2] format was chosen as a starting point for our research. An XML database is employed to store collections. For the information retrieval requests PF/Tijah, an XQuery extension made for this purpose, was used.

CLEF 2006 CL-SR is an ideal evaluation scenario for our research because: (i) The provided test collection existed in XML-like format, (ii) the collection was pre-segmented, (iii) each segment contained manual and automatic annotations, and (iv) these annotations could be combined in order to exploit their added value. CLEF 2006 CL-SR offers therefore a welcome framework for evaluating the preliminary implementation of our ideas described below. A second reason is that it links up to other work on spoken document retrieval for oral history collections done at the University of Twente, in particular to the CHoral project, which is focusing on speech retrieval techniques for Dutch cultural heritage collections. [3].

The rest of the paper is organized as follows. In section 2 the PF/Tijah module, its application to the retrieval task together with the cross-lingual application are presented. Section 3 presents our results on the given retrieval topics. We end the paper by giving a conclusion and future work in Section 4.

2 PF/Tijah: Text Search in XQuery

PF/Tijah is a research project run by the University of Twente with the goal to create an extendable search system. It is implemented as a module of the PathFinder XQuery (PF) execution system [4] and uses the Tijah XML Information Retrieval system [5]. PF/Tijah is part of the XML Database Management System MonetDB/XQuery. The whole system is an open source package developed in cooperation with CWI Amsterdam and the University of Munich. It is available from SourceForge.[1] The rest of this section is structured as follows: In Subsection 2.1 we describe the PF/Tijah system. The next Subsection 2.2 gives insights on application of the PF/Tijah system to perform the given tasks on the CLEF-2006 CL-SR collection. The last Subsection, 2.3, gives details on the cross-language aspects.

2.1 Features of PF/Tijah

PF/Tijah supports retrieving arbitrary parts of XML data, unlike traditional information retrieval systems for which the notion of a *document* needs to be defined up front by the application developer. For instance, if the data consists of MPEG-7 annotations, one can retrieve complete documents, but also only parts inside documents with no need to adapt the index or any other part of the system.

Information retrieval queries are specified in PF/Tijah through a user-defined function in XQuery that takes a NEXI query as its argument. NEXI [6] stands for *Narrowed Extended XPath*. It is narrowed since it only supports the descendant and the self axis steps (which selects the current context node - opposed to all nodes beneath it). "Extended" because of its special `about()` function that takes a sequence of nodes and ranks those by their estimated probability of relevance to the query consisting of a number of terms.

PF/Tijah supports extended result presentation by means of the underlying query language. For instance, when searching for document parts, it is easy to include any XML element in the collection in the result. This can be done in a declarative way (i.e., not focusing on how-to include something but on what to include). This could be additional information on the author of the document or technical information such as the encoding. The result is gathered using XPath expressions and is returned by means of XQuery element constructions.

As a result of the integration of information retrieval functionality in XQuery, PF/Tijah information retrieval search/ranking can be combined with traditional database query techniques, including for instance joins of datasets on certain values.

For example, one could search for video parts in which a word is mentioned, that is listed in a *name* attribute in an XML document collection on actors. It is also possible to narrow down the results by constraining the *actor* element further by, say, the earliest movie he played in.

[1] http://sourceforge.net/projects/monetdb/

2.2 Querying the CLEF-2006 CL-SR Collection

For a description of the anatomy of the CLEF-2006 CL-SR collection please refer to [7]. PF/Tijah can be used to query this collection as follows. Considering topic 1133, *"The story of Varian Fry and the Emergency Rescue Committee who saved thousands in Marseille"*, a simple CLEF CL-SR query that searches the ASR transcript version from 2004 is presented in Fig. 1. To understand the query it is necessary to have some basic knowledge of XQuery as for instance described by Katz et al. [8]. In the query, we specify which elements to search (in this case `ASRTEXT2004A`) explicitly. It is easy to run queries on other elements (annotations), for instance the `SUMMARY` elements, without the need to re-index. Furthermore, queries using several elements can be done by combining multiple `about()` functions with `and` or `or` operators in the NEXI query.

```
let $c := doc("Segments.xml")
for $doc in tijah-query($c, "//DOC[about(.//ASRTEXT2004A,
   varian fry)];")
return $doc/DOCNO
```

Fig. 1. Simple PF/Tijah query

```
let $opt := <TijahOptions returnNumber="1000" algebraType="COARSE2"
  txtmodel_model="NLLR"/>
let $c := doc("Segments.xml")
(: for all topics :)
for $q in doc("EnglishQueries.xml")//top
  (: generate NEXI Query :)
  let $q_text := tijah-tokenize(exactly-one($q/title/text()))
  let $q_nexi := concat("//DOC[about(.//ASRTEXT2004A,", $q_text, ")];")
  (: start query on the document node :)
  let $result := tijah-query-id($opt, $c, $q_nexi)
  (: for each result node print the standardized result :)
  for $doc at $rank in tijah-nodes($result)
  return
    string-join(($q/num/text(), "Q0", $doc/DOCNO/text(), string($rank),
      string(tijah-score($result, $doc)), "pftijah"), " ")
```

Fig. 2. PF/Tijah query that completely specifies a CLEF run

Interestingly, we can specify one complete CLEF CL-SR run in one huge XQuery statement as shown in Fig. 2. The query takes every topic from the CLEF topics file, creates a NEXI query from the topic title, runs each query on

the collection, and produces the official CLEF output format. The <TijahOptions /> element contains options for processing the query, such as the retrieval model that is supposed to be used. More about PF/Tijah can be found in [9].

2.3 Cross-Lingual IR

For cross-language retrieval Dutch topics [7] were used as queries on the English collection. For this purpose all Dutch queries were fully automatically translated into English and then processed with the procedure described above.

As a machine translation (MT) system we used the free web translation service *freetranslation.com*[2], which is based on the SDL Enterprise Translation Server, a hybrid phrase based statistical and example based machine translation system. There was no pre- or post-processing carried out. The results using *freetranslation.com* seemed to be performing well, though no formal evaluation of the translation quality was carried out. Since online MT systems like this are being changed regularly, it will be difficult to repeat the achieved results. This is one of the reasons we are considering building our own statistical MT system for a future CLEF evaluation.

3 Experimental Results

We tested the out-of-the-box performance of PF/Tijah with provided, unmodified metadata. Table 1 shows the experimental results of 9 experiments. One experiment is the obligatory ASR run using relatively long queries consisting of of topic title and topic description (denoted as 'TD' in the table). The other eight runs were done using short queries with only the topic title ('T' in the table). Five out of nine runs use the original English topics ('EN' in the table). The four remaining runs use the manually translated Dutch topics ('NL' in the table) to search the English annotations. Runs were done on different annotations: on the ASR transcripts version 2004 ('ASR04'), on the manual keywords ('MK') and/or on the manual summaries ('SUM').

Table 1 reports the LEF 2006 mean average precision results for 63 training topics and 33 test topics. The first five runs were officially submitted the other ones were self-assessed. The ASR-only results are significantly worse than any of the runs that use manual annotations (manual keywords or summaries of interview parts). Best results are obtained by combining the manual keywords and summaries. Interestingly, on the test topics, the difference between manual keywords only (UTmkEN) and manual keywords plus ASR (UTasr04mkEN) is statistically significant at the 5% level according to a paired sign test.

4 Conclusions and Future Work

Expectedly, retrieval results using ASR transcripts are far off from the results of manual annotation. For example, manually annotated summaries combined with

[2] http://www.freetranslation.com/

Table 1. Mean average precision (map) on train and test queries (slightly differs from official values due to fixed bugs)

Run name	Lang.	Query	Annotations	Train (map)	Test (map)
UTasr04aEN-TD	EN	TD	ASR04	0.069	0.047
UTasr04aEN	EN	T	ASR04	0.058	0.052
UTasr04aNl2	NL	T	ASR04	0.053	0.039
UTsummkENor	EN	T	MK, SUM	0.246	0.203
UTsummkNl2or	NL	T	MK, SUM	0.210	0.169
UTmkEN	EN	T	MK	0.201	0.151
UTmkNL2	NL	T	MK	0.163	0.119
UTasr04mkEN	EN	T	ASR04, MK	0.218	0.162
UTasr04mkNL2	NL	T	ASR04, MK	0.177	0.129

manual keywords have 0.2 mean average precision compared to 0.05 for the ASR transcripts. Taking costs and the available resources for the annotation process into account, manual annotation often turns out to be infeasible due to limited resources. Therefore, if no manual annotations are available – which is true for quite a number of audiovisual collections – automatic annotation at least allows a specific document to be considered if it contains relevant information to the user. In addition, experiments show that the combination of ASR and MK annotations improve the precision of the results significantly, compared to sole use of MK. Therefore we expect that the combination of manual and automatically generated annotations will be beneficial in general.

As it was stated in the introduction, using multiple annotations gives rise to several unsolved, important questions. The experiments described in this paper have been a first exploration of these issues using a well-defined spoken document retrieval evaluation corpus. We aim to continue this line of research by investigating (i) how annotation layers can most efficiently be stored, (ii) how retrieval scores could best be combined, and (iii) how document segmentation should be dealt with.

The storage structure used for the annotations was the predefined format (transformed to valid XML) of the CLEF-2006 CL-SR collection. All annotations were aligned after one segmentation scheme: parts of interviews on one coherent topic. This way all annotations nicely fitted into a rather flat hierarchical structure. In other also realistic settings segmentations could be more deeply nested or overlapping. For example, an automatically detected emotion could span multiple parts of an interview. This leaves the question open of how to efficiently store multiple annotations. Here, existing standards for data structures for the storage of annotations, like MPEG-7, have to be considered and evaluated.

The retrieval of information based on manual and automatic annotations was done so far using an "or" operator. This assumes that both annotation types contribute equally to the relevance of a segment. It is however more realistic to assume some kind of ordering or weighting. For example, manual keyword

annotations might receive a larger weight than findings in the ASR transcript. How to solve the problem of possible overlapping annotations during the combination of retrieval results on different annotations is still to be settled.

With possibly huge amounts of content in one single multimedia document a retrieval system has to be able to select only certain parts which are presented to the user as being relevant. Because annotations may segment a document in multiple ways, the issue of which part to return as relevant needs to be addressed in more detail.

Acknowledgements

We are grateful to the research programmes that made this work possible: Djoerd Hiemstra and Roeland Ordelman were funded by the Dutch BSIK project MultimediaN[3]. Robin Aly was funded by the Centre of Telematics and Information Technology's SRO-NICE programme[4]. Laurens van der Werff was funded by the Dutch NWO CATCH project Choral[5]. Many thanks to Arthur van Bunningen, Sander Evers, Robert de Groote, Ronald Poppe, Ingo Wassink, Dennis Reidsma, Dolf Trieschnigg, Lynn Packwood, Wim Fikkert, Jan Kuper, Dennis Hofs, Ivo Swartjes, Rieks op den Akker, Harold van Heerde, Boris van Schooten, Mariet Theune, Frits Ordelman, Charlotte Bijron, Sander Timmerman and Paul de Groot for translating the English topics to Dutch.

References

1. MPEG-Systems-Group: ISO/MPEG N4285, Text of ISO/IEC Final Draft International Standard 15938-1 Information Technology - Multimedia Content Description Interface - Part 1 Systems. ISO (2001)
2. Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E.: The extensible markup language (xml) 1.0 3rd edn. Technical report, W3C (2004)
3. Ordelman, R., de, F.J., van A.J.H.: The role of automated speech and audio analysis in semantic multimedia annotation. In: Proceedings of the International Conference on Visual Information Engineering (VIE2006), Bangalore, India (September 2006)
4. Boncz, P., Grust, T., van Keulen, M., Manegold, S., Rittinger, J., Teubner, J.: MonetDB/XQuery: A fast XQuery processor powered by a relational engine. In: Proceedings of the ACM SIGMOD Conference on Management of Data, ACM, New York (2006)
5. List, J., Mihajlovic, V., Ramirez, G., de Vries, A., Hiemstra, D., Blok, H.: Tijah: Embracing IR methods in XML databases. Information Retrieval Journal 8(4), 547–570 (2005)
6. O'Keefe, R., Trotman, A.: The simplest query language that could possibly work. In: Proceedings of the 2nd Initiative for the Evaluation of XML Retrieval (INEX) (2004)

[3] http://www.multimedian.nl/
[4] http://www.ctit.utwente.nl/research/sro/nice/
[5] http://hmi.ewi.utwente.nl/project/CHoral

7. Peters, C., Claugh, P., Gey, F., Karlgen, J., Magnini, B., de Rijke, D.W.D.O., Stempfhuber, M. (eds.): Evaluation of Multilingual and Multi-modal Information Retrieval Seventh Workshop of the Cross-Language Evaluation Forum CLEF 2006, September 2006, Alicante, Spain. Springer, Heidelberg (to appear, 2007)
8. Katz, H., Chamberlin, D., Draper, D., Fernandez, M., Kay, M., Robie, J., Simeon, M.R.J., Tivy, J., Wadler, P.: XQuery from the Experts: A Guide to the W3c XML Query Language. Addison-Wesley, Reading (2003)
9. Hiemstra, D., Rode, H., van Os, R., Flokstra, J.: PFTijah: text search in an XML database system. In: Proceedings of 2nd International Workshop on Open Source Information Retrieval (OSIR) (2006)