A Spoken Document Retrieval System for TV Broadcast News in Spanish and Basque

Sistema de recuperación de noticias de televisión en castellano y euskera

University of the Basque Country, UPV/EHU
GTTS, Department of Electricity and Electronics
amparo.varona@ehu.es

Resumen: El sistema de indexado y búsqueda de contenidos multimedia que se presenta en este trabajo (Hearch) es un buscador de aspecto convencional pero con la capacidad de devolver segmentos de video gracias a la transcripción automática de sus contenidos de voz. El sistema consta de un back-end que capta, procesa e indexa los recursos, y de un front-end que permite realizar búsquedas y configurar y monitorizar el funcionamiento de los distintos módulos, mediante una interfaz web. Actualmente se encuentra operativa una versión de la herramienta que trabaja frente a repositorios de noticias en castellano y euskera (http://gtts.ehu.es/Hearch/). Para evaluar el rendimiento del sistema se dispone de 6 programas de noticias en castellano y 7 en euskera. Puesto que el módulo de Reconocimiento Automático del Habla introduce bastantes errores, se ha propuesto y evaluado una aproximación basada en añadir términos afines a los de la pregunta para ampliar los resultados proporcionados por el sistema. Como resultado se obtiene una pequeña mejora del rendimiento.

Palabras clave: Recuperación de recursos multimedia, Reconocimiento Automático del Habla

Abstract: This paper presents a spoken document retrieval system (Hearch) looking like a conventional search tool, which retrieves audio/video segments based on the automatic transcription of speech contents. The system consists of a back-end that captures, processes and indexes audio/video resources, and a front-end that allows to search contents, configure various modules and display performance statistics through a web interface. An early version of this tool is available (http://gtts.ehu.es/Hearch/), which searches and retrieves segments on TV broadcast news repositories in Spanish and Basque. To evaluate the performance of the system, six manually transcribed TV broadcast news in Spanish and seven in Basque have been used. An approach based on extending the query with the so called friendly terms has been proposed and evaluated, attempting to minimize the effect of errors introduced by the Automatic Speech Recognition module. This approach led to slight performance improvements.

Keywords: Spoken Document Retrieval, Automatic Speech Recognition

1. Introduction

The fast growth of multimedia (audio/video) contents available in internet makes it necessary searching into such kind of resources. Spoken Document Retrieval (SDR) systems look like conventional search tools (such as Google, Bing, etc.): users introduce query terms and the system outputs audio segments where those terms have been pronounced, sorted according to a given measure of relevance. To achieve that goal, speech must be converted to text or some text describing resource contents must be used. Some systems index and categorize videos using the titles or brief descriptions (metadata) that accompany these resources, such as the tags...
provided by users. These texts are just short descriptions, shallow categorizations or partial transcriptions, so the resulting index is very coarse and the search cannot focus on specific items. Products developed by companies, such as VideoSurf\(^1\), Delve Networks\(^2\) and Truveo\(^3\) use this kind of information.

Clearly, search engines can take a key advantage from using Speech Technologies. The use of Automatic Speech Recognition (ASR) allows to transcribe spoken documents and then text-based Natural Language Processing (NLP) tools can be applied. But automatically generated transcripts still include a large number of errors, which must be taken into account in any information extraction process. In a first phase, several audio processing modules (audio segmentation, audio classification, language verification, speaker identification/diarization, speech recognition, etc.) must be sequentially applied to audio streams, and the information related to segmentation, audio type, language, speaker and the recognized transcription stored for further processing. Text processing tools can be applied to enrich the transcription with morphosyntactic information. Finally, the resulting text must be used to build an index to allow efficient information retrieval (Makhoul et al., 2000).

In the last years, some systems have been already developed applying Large-Vocabulary Continuous Speech Recognition (LVCSR). SpeechBot (Thong et al., 2002) was an experimental web-based tool from HP Labs that used speech recognition to create searchable keyword transcripts from thousands of hours of audio content. SpeechFind (Hansen, 2005) is a spoken document retrieval system developed by the Center for Robust Speech Systems at the University of Texas at Dallas which was used to transcribe the National Gallery of Spoken Words, which covers up to 60000 hours of USA historic recordings from the last 110 years. The Massachusetts Institute of Technology (MIT) system (Glass et al., 2007) dealt with improving the access to on-line audio/visual recordings of academic lectures. The system developed at NTT (Ohtsuki et al., 2006) indexes multimedia contents in Japanese. The ASEKS system (Ye et al., 2006) uses keyword spotting technology for indexing spoken documents in Chinese.

But LVCSR systems do not provide universal coverage: some query terms may not appear in the vocabulary. To solve the lack of coverage, some proposals such as those of Nexidia (Clements and Gavalda, 2007) and IBM (Mamou and Ramabhadran, 2008) are based on spoken term detection. First, the input speech is processed to produce a phonetic decoding, which is searched in a second stage to find the query terms. The main disadvantage of this approach is that search must be performed for each query, and each audio file, thus introducing considerable delays.

Other systems combine metadata information with ASR outputs. For example, the Google Speech Research Group (Alberti et al., 2009) developed an audio indexing system for the videos of YouTube corresponding to the 2008 presidential election race in the United States. Another example is a Korean spoken document retrieval system for Lecture Search (Lee and Lee, 2008).

Currently, the best positioned companies in the video search market are Blinkx\(^4\), Ramp\(^5\), Nexidia\(^6\), TVEyes\(^7\), Gaudi\(^8\) (Google Audio Indexer appeared in the third quarter of 2008) and MAVIS\(^9\) (Microsoft Research Audio Video Indexing System). It might seem, by the reported performances and the appearance of their interfaces, that these tools have solved the problem of indexing and searching multimedia contents. However, this is a misperception, since these tools focus on a very specific type of resources to optimize performance.

In this paper, a Spoken Document Retrieval System (Hearch) which deals with resources in both Spanish and Basque languages is presented and evaluated. The system retrieves audio/video segments based on the automatic transcription of speech contents. It consists of a back-end that captures, processes and indexes audio/video resources, and a front-end that allows to search contents, configure various modules and display performance statistics through a web interfa-

\(^1\)http://www.videosurf.com/
\(^2\)http://www.delvenetworks.com/
\(^3\)http://www.truveo.com/
\(^4\)http://www.blinkx.com/
\(^5\)http://www.ramp.com/
\(^6\)http://www.nexidia.com/
\(^7\)http://www.tveyses.com/
\(^8\)http://labs.google.com/gaudi
ce. An early version of this tool is available (http://gtts.ehu.es/Hearch/) working on
1TB of TV broadcast news in Spanish and Basque obtained from Euskal Irrati Telebista
(EITB). A preliminary evaluation has been carried out on six manually transcribed Spa-
nish and seven Basque TV broadcast news and two sets of query terms.

The rest of the paper is organized as fol-
lows. Section 2 describes the spoken docu-
ment retrieval system, including modules re-
lated to audio processing (audio segmenta-
tion, language identification, speaker identi-
fication, automatic speech recognition), the
NLP tools for text processing, the indexer,
the search engine and the user interface. Sec-
tion 3 describes the databases used in the eva-
luation. In Section 4 the Automatic Speech
Recogntion module (the key piece of the sys-
tem) is evaluated independently. In Section 5
the whole spoken document retrieval system
is evaluated. Finally, conclusions are given in
Section 6.

2. The Spoken Document
Retrieval System

The SDR system (Hearch) is based on
a simple and efficient architecture, which
allows to replace or integrate new modules
in a easy and elegant way (see Figure 1).
In the back-end the information obtained at
each step is incrementally stored in an XML
resource descriptor (Bordel et al., 2009). Au-
dio signals are segmented and classified, the
speech segments are transcribed and the re-
sulting sequences of words are lemmatized
yielding an enriched XML document. This
document is taken as input by the indexer to
update an index database, which is the core
data structure of the system. In the front-end,
each time a user makes a query, the query is
lemmatized and then the search engine tra-
verses the index database to find matching
resources. A web interface allows users to ac-
cess Hearch from remote locations, to enter
queries and to receive search results.

2.1. The back-end

All the operations described in this section
are performed once in off-line mode, before
user smake their queries.

2.1.1. The crawler

Copies of the original resources are kept
locally, and the audio streams are converted
into PCM format for further processing. An

2.1.2. Audio processing

Audio is processed in several steps. At
each step, the XML document is enriched with
information specific to a knowledge le-
vel.

– Audio Segmentation. This task con-

sists of dividing a continuous audio stream
into acoustically homogeneous regions called
segments. There are robust and unsupervised
 techniques for doing it. (Kiranyaz, Qureshi,
and Gabbouj, 2006). In particular, the identi-

fication of speech and non-speech segments is
a key step. If non-speech segments are exclu-
ded from recognition, not only computation
time is saved in ASR, but also better trans-
criptions are obtained. Small interruptions li-
ekoughs and other noises produced by the
speaker are admitted inside of a speech seg-
ment (Rodriguez-Fuentes et al., 2010).

– Language Identification. In multilin-
gual systems, identifying the language spo-
ken in each segment is necessary for the
ASR system to use adequate acoustic and
syntactic models. In a previous work, lan-
guage recognition results obtained for the
four official Spanish languages (Spanish, Ca-
talan, Basque and Galician) were presented.
Best performance was achieved with the fu-
sion of an acoustic system and 6 phonotac-
tic subsystems. Acoustic systems take infor-
mation from the spectral characteristics of
the audio signal, whereas phonotactic sys-
tems use sequences of phones produced by se-
veral acoustic-phonetic decoders (Varona et
al., 2010).

– Speaker Identification. This task con-

sists in classifying the speech segments in
terms of speakers, which has always a pos-
tive impact on the accuracy of the ASR sys-
tem, by applying model adaptation techni-
ques (unsupervised clustering of similar voi-
ces, Bayesian adaptation, etc.) (Diez et al.,
2011). Moreover, if speaker profiles were avai-

able beforehand, then speaker turns could be
identified, which is interesting from the point
of view of indexing, since users might be in-
terested in finding the segments correspon-
ding to a given speaker.

– Automatic Speech Recognition. Speech recognition is the process of con-
verting an acoustic signal to a sequence of
words. ASR systems are invariably based on
the well-known Bayes rule (Jelinek, 1999),
i.e the recognizer looks for the most likely word sequence according to previously estimated acoustic models (typically, hidden Markov models) and language models (typically, n-grams). Acoustic models estimate the frequency distributions of sounds over time, and language models estimate the frequency of word sequences. Specific acoustic and language models are trained for each language. This module is currently implemented by means of Sautrela\(^{10}\) (Penagarikano and Bordel, 2005), an open-software package for speech processing, entirely developed using Java. The Hearch architecture also supports the inclusion of other components, such as the recognition engine of HTK (Young, 2006).

- **Text processing.** The morphosyntactic analysis of the recognized sequences of words allows us to know the lemma of each word, which helps to index and search inflected forms. To analyse words in Spanish, the well-known FreeLing package (Atsierias et al., 2006) is used. Basque is a deeply agglutinative language and specific tools are applied to carry out lemmatization (Aduriz et al., 1998).

### 2.1.3. The indexer

The collection of enriched XML documents is taken as input to the indexer tool, which creates a hierarchized structure of term references. The indexing process uses a finite set of terms: the vocabulary (predefined) of the ASR module \(T = t_1, t_2, \ldots, t_{|T|}\). Each segment in the document is treated as a subset of terms (the order is not taken into account). To lessen computational costs, terms supposed to be useless to represent document content (the so called stopwords, such as articles, conjunctions, pronouns, etc.) have been eliminated. Each segment is represented as a vector, and a weight \(w_{jk} > 0\) is associated with each term \(k\) of a segment \(d_j = (w_{j1}, w_{j2}, \ldots, w_{jk}, \ldots, w_{j|V|})\). For a term that does not appear in segment \(d_j\), \(w_{jk} = 0\). Usually, each term weight is computed based on variations of the \(tf\) or \(tf - idf\) schemes\(^{11}\) (Mills et al., 2000).

In this work, the indexer is implemented by means of Apache Lucene (Hatcher, Gospodnetic, and M., 2010), a high-performance full-featured text search engine library written entirely in Java. The index structure, which contains location information for each term, is dynamically updated each time a new XML resource is added to the system.

\(^{10}\)http://sautrela.org/

\(^{11}\)tf is the Term’s Frequency, defined as the number of times that term \(k\) appears in the document \(d_j\). The Inverse Document Frequency idf gives more weight to terms occurring in few documents.
2.2. The front-end

The information retrieval module (the search engine) and the web interface work on-line, by processing user queries and presenting a ranked list of results.

2.2.1. The search engine.

The information retrieval process begins when the user formulates a query. Again text processing tools (lemmatization) are applied to represent the query \( q \) in the same way as segments \( d_j \), i.e. as a feature vector. The system retrieves those segments matching the query terms in the index database, and ranks them according to some predefined matching measure (Frakes and Baeza-Yates, 1992). The standard relevance ranking of segments of Apache Lucene has been modified in order to penalize short segments, so we assign the score taking into account the square root of the number of terms (\( \text{NumTerm} \)) contained in the segment in the following way:

\[
\text{Score} = C \sum tf \cdot idf^2 \sqrt{\text{NumTerm}}
\]

where \( C \) is a fraction of the number of query terms found in the segment.

2.2.2. The user interface

The user interacts with the SDR system from a remote computer through internet, using a simple navigator (see Figure 2). A web server is accepting queries and sending them to the search engine, which returns a rank of matching segments. Finally, the system interface composes and presents successive HTML pages showing a list of matching items, with information about the resource name, location and size, segment boundaries (time stamps), links (thumbnails) to cached copies of the original multimedia resources, transcription excerpts which link to the full recognized transcriptions, and relevant metadata (topics, speakers, etc.) stored in the XML resource descriptors.

3. Evaluation databases

An early version of the SDR system is available at http://gtts.ehu.es/Hearch/, working on 1TB of TV news in Spanish and Basque recorded from EITB. For evaluation purposes, two small databases have been created in both Spanish and Basque languages:

- For Spanish, a collection of 6 TV news (4h 47min stored in 1GB) is available, including 3918 segments, manually segmented and transcribed, with 7779 different words (terms) in the vocabulary.
- For Basque, a collection of 7 TV news (4h, stored in 500MB) is available, including 3391 segments, manually segmented and transcribed, with 8093 different words (terms) in the vocabulary.

4. ASR Evaluation

The automatic speech recognition module is the key piece of the system. Acoustic and Language Models have been trained on large acoustic and text databases and the evaluation has been carried out on a small but independent set of speech segments taken from TV broadcast news.

- **Acoustic features.** Audio signals are stored in PCM format at 16KHz, 16 bit per sample. Acoustic features consisted of 12 Mel Frequency Cepstral Coefficients (MFCC) together with log-energy, calculated every 10ms in sliding Hamming windows of 25ms (first order preemphasis with 0.97 coefficient, and a 26 channel filterbank were applied). Delta and double delta coefficients were also computed, resulting in a 39-component feature vector.

- **Acoustic modeling.** The acoustic modeling is based on left-to-right continuous Hidden Markov Models with three looped states and 64 Gaussian mixtures per state. The phone inventory consisted of 23 phone units for Spanish and 26 phone units for Basque. Both languages share the five vowels but Basque has more fricative and affricate sounds.

For Spanish, the well-known Albayzin database (Moreno et al., 1993) was used to es-
timate acoustic models. It consists of 6800 read sentences from 204 speakers: 4800 from 164 speakers for training and 2000 from 40 speakers for testing.

For Basque, the acoustic database AS3200 (Aditu) (Basque-Government, 2005) was used. Recordings were made in an office environment, following the specifications of SPEECON 12 (Siemund et al., 2000). In this work, only the subset of read sentences was used, consisting of 8298 sentences from 215 speakers: 5346 from 140 speakers for training and 2952 from 75 speakers for testing.

- **Language modeling.** Two databases were used to estimate the language models.

  - Spanish ML contains text news in Spanish taken from the internet from 2005 to 2008 including various topics such as economy, society, sports, opinions, politics, etc. It contains 1,790,654 sentences and 50,862,981 words. The size of the vocabulary for this database is 302,430.


The SRI Language Modeling Toolkit SRILM (Stolcke, 2002) was used to estimate n-grams Language Models (LM). Classical Witten-Bell smoothing was applied to tri-gram LMs. Three different vocabulary sizes (dictionaries) were evaluated: (1) 5K words, including the 5000 most likely words in the whole text database; (2) 20K words, including the 20000 most likely words in the whole text database; and (3) closed dictionary, which contains all the words in the 6 TV news in Spanish (7759) and the 7 TV news in Basque (8093) (see Section 3).

- **Test sets.** The experiments were carried out on a selection of segments manually extracted from the databases described in Section 3. Segments were classified into three groups according to their audio characteristics: (1) presenter: audio signals in this group are clean because they were recorded in a TV studio and the presenter was reading a script; (2) reporter: this group contains audio signals recorded by reporters on the streets, so the speech is less formal than in previous ones and probably contains background noise; and (3) spontaneous: this group contains spontaneous speech (commonly produced in interviews) so the signals will probably contain a lot of disfluencies. Table 1 shows the number of segments, the number of words and the vocabulary size for each collection of segments.

<table>
<thead>
<tr>
<th></th>
<th>segments</th>
<th>words</th>
<th>vocab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spanish</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>presenter</td>
<td>180</td>
<td>5045</td>
<td>1797</td>
</tr>
<tr>
<td>reporter</td>
<td>180</td>
<td>4482</td>
<td>1655</td>
</tr>
<tr>
<td>spontaneous</td>
<td>96</td>
<td>2594</td>
<td>914</td>
</tr>
<tr>
<td>total</td>
<td>456</td>
<td>12121</td>
<td>4366</td>
</tr>
<tr>
<td>Basque</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>presenter</td>
<td>180</td>
<td>2971</td>
<td>1707</td>
</tr>
<tr>
<td>reporter</td>
<td>180</td>
<td>2885</td>
<td>1626</td>
</tr>
<tr>
<td>spontaneous</td>
<td>90</td>
<td>1569</td>
<td>807</td>
</tr>
<tr>
<td>total</td>
<td>450</td>
<td>7425</td>
<td>4140</td>
</tr>
</tbody>
</table>

- **Results.** Test segments (see Table 1) were processed by the ASR decoder which outputs sequences of words (without lemmatization). Table 2 shows the ASR performance for the three sets of sentences (presenter, reporter and spontaneous) and the three LMs (5K, 20K and closed). Classical Word Accuracies 13 were calculated. Clearly, Spanish decoders run better than Basque decoders when LM of 5K and 20K were considered. However, for closed-set LMs best performances were obtained for Basque. That is because Basque is a highly inflected language in terms of both nouns and verbs, with 17 cases, thus the 5K or 20K LMs were less general than their counterparts for Spanish.

Performance was also measured when lemmatization was applied both to the recognized sequences of words and user queries. Table 3 shows Lemma Accuracy 14. Performance improved, specially when considering vocabularies of 5K and 20K words, both for Spanish and for Basque.

## 5. Information Retrieval Results

- **Choice of query terms.** We have manually defined four groups of queries, three of them according to frequency: 15 terms that appear frequently in the vocabulary of

---

12SPEECON, launched in February 2000, focused on collecting linguistic data in different European languages for speech recognition applications.

13Word Accuracy takes into account the Levenshtein distance in a dynamic string alignment: \( \text{Acc} = (C - I)/N \), where \( C \) is the number of correctly recognized words, \( I \) is the number of inserted words, and \( N \) is the number of words in the reference transcription.

14Lemma Accuracy is defined the same way as Word Accuracy but considering lemmas instead of words.
the TV news, 25 terms that appear less frequently and 25 unusual terms. The forth group contains 30 queries composed of pairs of terms, with some kind of relationship between them, from the three previous groups. Eventually, we decided to join the first three groups into one, since performance differences observed for the three groups of frequencies were not significant. Thus, we considered two groups of queries, one composed of 65 one-term queries and another composed of 30 two-term queries.

- **Expanding the query with friendly terms.** Adding friendly terms is an attempt to improve the performance of the system by searching for terms with high probability of appearing in the same sentence than query terms. To create a list of friendly terms, it is necessary to calculate for every sentence in a big set of sentences (for example the Spanish ML and Basque ML databases) the conditional probability of each term given another term. Given two terms A and B, the joint probability $P(A, B)$ is calculated as:

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A) \quad (2)$$

This results in the well-known Bayes formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (3)$$

To avoid that less common terms appear as good candidates, two conditions are imposed: (1) $P(B|A) > P(B)$; and (2) $P(B) > \text{threshold}$. This threshold should be close to the inverse of the number of different terms in the text. In this work, $\text{threshold} = 0.001$.

- **Evaluation measures.** The Search Quality Benchmarking package of Lucene was used to obtain all the search results given a query, and TREC Eval\(^{15}\) was used to estimate the similarity between the segment and the query, by means of the Mean Average Precision (MAP) measure:

$$\text{MAP} = \frac{\sum_{q=1}^{Q} \text{AveP}(q)}{Q} \quad (4)$$

MAP is the mean of the average precision scores for each query $q \in Q$. The average precision is defined as follows:

$$\text{AveP} = \frac{\sum_{r=1}^{N} (P(r) \times \text{rel}(r))}{\text{Number Relevant Segments}} \quad (5)$$

where $r$ is the rank, $N$ the number of retrieved segments and $P(r)$ the precision at a given rank $r$. If the segment at rank $r$ is non-relevant, the binary function $\text{rel}(r) = 0$. Sometimes, this metric is also referred to geometrically as the area under the Precision-Recall curve. The Percentage of Relevant Retrieved (PRR) segments is also used as evaluation measure in this work.

- **Results.** The SDR system was applied on the set of segments described in Section 3, that is, 3918 for Spanish and 3391 for Basque. Table 4 shows MAP and PRR performances for the two sets of one-term and two-term queries mentioned above. Performance for Basque was worse than for Spanish, maybe because ASR results (the input to the IR system) were also worse. In Spanish the use of two-term queries yielded better results.

Table 5 shows MAP and PRR performance for the same sets when adding friendly terms. Experiments with different number of

| Table 2: Word Accuracies for different ML (5K, 20K and closed-set) considering different types of segments (presenter, reporter and spontaneous). |
|----------------------------------|---|---|---|
|                                | 5K | 20K | closed |
| **Spanish**                    |    |    |      |
| presenter                      | 64.74 \% | 73.05 \% | 86.7 \% |
| reporter                       | 68.45 \% | 77.71 \% | 89.47 \% |
| spontaneous                    | 43.56 \% | 46.11 \% | 58.87 \% |
| mean                           | 58.91 \% | 65.62 \% | 78.34 \% |
| **Basque**                     |    |    |      |
| presenter                      | 49.85 \% | 59.81 \% | 91.89 \% |
| reporter                       | 49.57 \% | 60.67 \% | 92.65 \% |
| spontaneous                    | 27.68 \% | 28.51 \% | 60.97 \% |
| mean                           | 42.36 \% | 49.66 \% | 81.83 \% |

| Table 3: Lemma Accuracies for different ML (5K, 20K and closed-set) considering different types of segments (presenter, reporter and spontaneous). |
|----------------------------------|---|---|---|
|                                | 5K | 20K | closed |
| **Spanish**                    |    |    |      |
| presenter                      | 67.76 \% | 75.62 \% | 87.18 \% |
| reporter                       | 71.46 \% | 79.67 \% | 90.00 \% |
| spontaneous                    | 46.26 \% | 48.77 \% | 59.91 \% |
| mean                           | 61.82 \% | 68.02 \% | 79.03 \% |
| **Basque**                     |    |    |      |
| presenter                      | 56.58 \% | 65.57 \% | 92.53 \% |
| reporter                       | 56.26 \% | 66.37 \% | 93.21 \% |
| spontaneous                    | 34.75 \% | 35.46 \% | 62.82 \% |
| mean                           | 49.19 \% | 55.80 \% | 82.85 \% |

\(^{15}\)Trec_eval is a standard tool to evaluate TREC (Text Retrieval Conference) results using the standard NIST evaluation procedures. It was written by Chris Buckley and it is available from the TREC website at trec.nist.gov/trec_eval/.
friendly terms were carried out. Table 5 only shows best results: (1) in the case of one-term queries, when 3 friendly terms were added; and (2) in the case of two-term queries, when 2 friendly terms were added (one per original term). Note that performance improved in all cases but not very significantly.

Table 5: MAP and PRR performance when (1) 3 friendly terms were added to one-term queries and (2) 2 friendly terms were added (one per original term) to two-term queries, for different LM (5K, 20K, closed-set).

<table>
<thead>
<tr>
<th></th>
<th>one-term queries</th>
<th></th>
<th>two-term queries</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP</td>
<td>PRR</td>
<td>MAP</td>
<td>PRR</td>
</tr>
<tr>
<td>Spanish</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5K</td>
<td>0.4687</td>
<td>64.23 %</td>
<td>0.5316</td>
<td>70.43 %</td>
</tr>
<tr>
<td>20K</td>
<td>0.5015</td>
<td>63.81 %</td>
<td>0.5579</td>
<td>68.07 %</td>
</tr>
<tr>
<td>closed</td>
<td>0.6562</td>
<td>72.96 %</td>
<td>0.6672</td>
<td>77.50 %</td>
</tr>
<tr>
<td>Basque</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5K</td>
<td>0.2727</td>
<td>41.00 %</td>
<td>0.2841</td>
<td>42.75 %</td>
</tr>
<tr>
<td>20K</td>
<td>0.3459</td>
<td>42.36 %</td>
<td>0.3350</td>
<td>43.26 %</td>
</tr>
<tr>
<td>closed</td>
<td>0.6780</td>
<td>71.11 %</td>
<td>0.6294</td>
<td>69.74 %</td>
</tr>
</tbody>
</table>

6. Conclusions

In this paper, a Spoken Document Retrieval system (Hearch) has been described and evaluated. In the back-end an XML resource descriptor is created for each multimedia file, and audio processing is carried out in four steps: audio segmentation, language identification, speaker identification (optional), automatic speech recognition (ASR), yielding an enriched XML document. ASR transcriptions are further processed by NLP tools to get sequences of lemmas (instead of words), which is suitable for highly inflected languages (such as Basque). The enriched XML document is taken as input by the indexer to update an index database, which is the core data structure of the system. In the front-end, each time a user makes a query the search engine traverses the index database to find the matching resources. A web interface allows users to access Hearch from remote locations, to enter queries and to receive search results.

To evaluate the performance of the system a small database was available, consisting of 6 Spanish and 7 Basque TV broadcast news, all of them manually segmented and transcribed. Better results were obtained for Spanish, maybe because Basque is a highly inflected language, thus needing much more data to estimate reliable language models. Since the sequence of recognized words includes many errors, we have tried to improve system performance by extending the query with the so called friendly terms, which makes the system to retrieve many additional segments. This approach led to slight performance improvements for both Spanish and Basque.

References


