2009 Seventh International Workshop on Content-Based Multimedia Indexing

Content-Based Search in Multilingual Audiovisual Documents using the International Phonetic Alphabet

Georges Quénot, Tien Ping Tan, Le Viet Bac, Stéphane Ayache, Laurent Besacier and Philippe Mulhem
Laboratoire d'Informatique de Grenoble
BP 53, 38041 Grenoble Cedex 9, France
Georges.Quenot@imag.fr

Abstract

*We present in this paper an approach based on the use of the International Phonetic Alphabet (IPA) for content-based indexing and retrieval of multilingual audiovisual documents. The approach works even if the languages of the document are unknown. It has been validated in the context of the "Star Challenge" search engine competition organized by the Agency for Science, Technology and Research (A*STAR) of Singapore. Our approach includes the building of an IPA-based multilingual acoustic model and a dynamic programming based method for searching document segments by "IPA string spotting". Dynamic programming allows for retrieving the query string in the document string even with a significant transcription error rate at the phone level. The methods that we developed ranked us as first and third on the monolingual (English) search task, as fifth on the multilingual search task and as first on the multimodal (audio and image) search task.*

1. Introduction

Audiovisual databases quite often contain documents in several languages. This is the case for instance for Internet streaming archives. It often happens that the language used in a document is unknown and that a given document contains spoken utterances in different languages. This significantly complicates the content-based search within these archives. One possibility is to apply a series of language recognizers and then apply the appropriate transcribing tool but language detectors make errors and unknown languages may be encountered. Another approach is to transcribe the documents at the phonetic level using a subset of the International Phonetic Alphabet (IPA) regardless of the actually spoken language. Content-based search can then be done at the level of IPA strings. This approach was promoted by the Agency for Science, Technology and Research (A*STAR) of Singapore in the context of the "Star Challenge" that they organized between March and October 2008[1]. This challenge also addressed the problem of the search in video documents using the image only and using combined audio and image information.

The Star challenge is organized as a competition for multimedia search engines. It is a bit different in its spirit from the classical evaluation campaigns in the field such as those organized by NIST. It is really a competition with conditions close to real world applications, in particular concerning the timing aspects (much more constrained), and it is less oriented towards fine method or system performance measurement and comparison. The challenge consists in a series of three eliminative rounds addressing respectively audio, image and multimodal content-based search. The five best ranked teams after the three rounds were invited to participate to a final competition in Singapore in "live" conditions. The search task at the audio level comes in two variants: in the first one (AT1), the query is given as a phonetic string (that could be either typed as such by a user or come from a text to phone converter); in the second one (AT2), the query is given as a spoken utterance and has to be transcribed in the same way as the audio documents. We took this opportunity for developing and testing innovative approaches in content-base audio and multimodal search.

We describe in this paper the methods that we developed for this participation and how we tested them in the context of this challenge. The paper is organized as follows: in section, 2, we describe how we built multilingual acoustic models; in section 3 and 4, we describe two approaches that we used for the IPA-based search, the first one using dynamic programming and the second one using the vector space model; in section 5, we described the experiments that we carried out in the context of the Star Challenge and we present the obtained results.

2. Audio processing

2.1. General Approach to deal with multilingual documents

Since the languages in the audio track were not known in advance, we decided to work on a multilingual approach for transcribing the audio track. One idea could have been to

1. http://hlt.i2r.a-star.edu.sg/starchallenge

978-0-7695-3662-0/09 $25.00 © 2009 IEEE
DOI 10.1109/CBMI.2009.42

run in parallel several automatic speech recognition (ASR) systems in different languages but it is not realistic in the Star Challenge context where time and computation constraints are very challenging.

Thus, we propose a "lower level" approach where a multilingual phonetic recognizer (with phone n-grams used as language model) is applied on both the documents and the queries. This recognizer has the advantages of being in principle language independent and very fast. In practice, the models are built using inputs from a small number of languages, both at the phonetic and the language model levels. A true "phone independent" phonetic decoder is difficult to develop and would not have been optimal for the targeted types of languages.

2.2. Preprocessing and ASR

First, for each video, the audio track is segmented into short segments using a Bayesian Information Criterion (BIC) based segmenter (see [1] for more details). The resulting audio segments roughly correspond to speaker turns and have an average duration of about 10 seconds. ASR is then applied on the audio segments resulting from this initial stage. No music detection and removal is applied on the audio track.

For ASR, Sphinx speech recognition system[2] with Sphinx-3 decoder[3] from Carnegie Mellon University (CMU) was selected for transcribing automatically the audio documents (database) and queries in the 1st knockout round (monolingual voice search tasks) and during the "qualifying race" (multilingual voice/video search tasks). In fact, Sphinx-3 is a fast speech decoder, capable of decoding and transcribing speech documents in real time. This is achieved using conventional Viterbi search strategy and beam heuristics. In addition, it has a lexicon-tree search structure. Sphinx-3 uses the acoustic models created by SphinxTrain and accepts n-gram language models in binary format, which are converted from a standard ARPA n-gram model.

The front-end module is used to preprocess the raw speech sampled at 16 kHz with a 16-bit accuracy to Mel-Frequency Cepstral (MFC) feature vectors together with their first and second derivatives. This produces feature vectors with a total of 39 dimensions. SphinxTrain makes use of the feature vectors to create a continuous HMM acoustic model. Phones were used as the unit of HMM, each having three states with a left-to-right topology. Conversely, the n-gram language model (made of phone units in our case) was created using CMU statistical language modeling toolkit [2] and the SRILM toolkit [3].

2. http://www.speech.cs.cmu.edu/sphinx/
3. http://cmusphinx.sourceforge.net/sphinx3/doc/s3_overview.html

2.3. Monolingual task

For the monolingual (native and dialectal English) voice search tasks, the native English acoustic models were composed of 4000 tied-states, each consisting of a mixture of 16 diagonal-covariance Gaussian densities. They have been trained using 140 hours of 1996 and 1997 HUB-4 broadcast news training data [4] by Carnegie Melon University [5]. We adapted these native English models with a small set of dialectal English (South East Asia region) speech data by a supervised MAP adaptation. The HUB-4 language models[4] and the CMU (Carnegie Mellon University) Pronouncing Dictionary[5] which contains over 125,000 words were used.

2.4. Multilingual task

For the multilingual voice/video search tasks, since no information about the input languages was given a priori, we decided to build multilingual models from four languages: English, Mandarin, Vietnamese and Malay (corresponding, we hope, to a large coverage of what can be found in the Singapore area). The multilingual acoustic models are all context independent models trained independently with 16 Gaussians mixtures, except for English acoustic models. Two English acoustic models were actually used, namely the HUB-4 context dependent acoustic model with 8 Gaussian mixtures from CMU, which was trained from broadcast news, and a context independent acoustic model with 8 Gaussian mixtures, which was trained from WSJ0 read corpus [6]. Since the HUB4 acoustic model from CMU is a context dependent model, we extracted only the context independent states out. The Mandarin acoustic model was trained from CADCC corpus [7], the Vietnamese acoustic model was trained from VnSpeechCorpus [8] and the Malay acoustic model was trained from a Malay corpus courtesy of University Sains Malaysia. Table 1 shows some characteristics of the used corpora including the number of speakers and the number of hours of audio signal.

Corpora	Description	Spk.	Hours
HUB4	English, broadcast news	–	140
WSJ0	English, read speech	123	15
VN	Vietnamese	29	15
CADCC	Chinese	20	5
MSC	Malay	18	5

Table 1. Corpora used for the training of the IPA transcriber

For language modeling, a multilingual phone-based bigram model was trained from multilingual text corpora for 4 languages. The use of phone-based language model speed up very significantly the speech decoder (around 0.25 RT).

4. http://www.speech.cs.cmu.edu/sphinx/models/
5. http://www.speech.cs.cmu.edu/cgi-bin/cmudict

3. Dynamic Programming based search

The search is always done at the level of IPA strings regardless of whether the transcription was initially done at the word level or at the phone level and regardless of whether the query was made as an IPA string (AT1) or as a spoken utterance (AT2). In all cases, we have to match and score IPA representations of both the queries and the documents. We have chosen to do it by using a variant of a word spotting algorithm [9]. The main difference between the original word spotting algorithm and our "IPA string spotting" algorithm is that we replace the audio feature vectors (typically Mel Frequency Cepstral Coefficients or MFCCs) by the IPA symbols.

3.1. Minimization of the edit distance

Due to frequent transcription errors either in the documents for both tasks and/or in the queries for AT2, the search for the query string within the document strings must allow for an inexact match. Matches, either exact or inexact, must also be given a score so that the documents with the most exact matches can be ranked first. In order to allow for inexact match and to score them, we chose the "edit distance" between a query string and a substring of a document string. All the possible matches between a query string and all the existing substring of all the document strings are considered and, for each such match, a distance is computed by counting and penalizing all the insertions, deletions and substitutions between the phone composing the query string and the document substring. Figure 1 shows an example of edit distance computation.

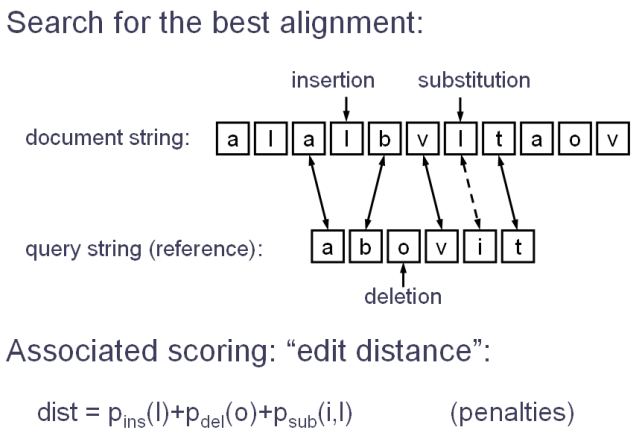

Figure 1. Scoring of a matching between a query string and a substring of a document string

3.2. Dynamic Programming

Dynamic programming is a way to solve the problem of finding and scoring the best alignment between a query string and a document substring in a computing time that is linear in both the query string length and the document string length.

The product matrix between the document string (horizontal) and the query string (vertical) is considered. A valid matching or alignment between the query string and a substring of the document string is an increasing path that joins the matrix bottom row to the matrix top row (Figure 2. The best alignment is the one that minimizes the edit distance along itself. The dynamic programming trick is to compute the best alignment by recurrence.

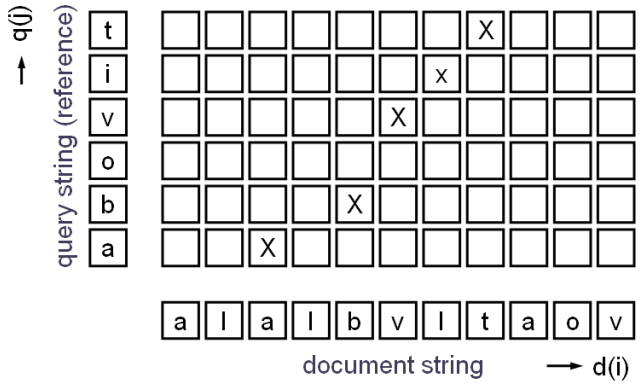

Figure 2. Matching as a path in the DP product matrix

If we consider the best edit distance $e(i, j)$ from the bottom row to the (i, j)-point, we have a recurrence equation on $e(i, j)$ since the best path arriving at the (i, j)-point must either:

- come from $(i - 2, j - 1)$ with an insertion penalty,
- come from $(i - 1, j - 2)$ with a deletion penalty or
- come from $(i - 1, j - 1)$ with a possible substitution penalty.

(unless one of these points is outside of the matrix).

$e(i, j)$ can be computed by recurrence in the whole matrix with an initialization to 0 on the bottom row and to "infinite" on the left column (excluding the bottom value). The actually used recurrence equation is given in Eq. 1. The c_{xx} are constants whose values are: $c_{ii} = c_{dd} = 2.0$, $c_{sn} = c_{sd} = 1.0$, $c_{si} = 0.5$ (normalization according to the query so that all alignments have the same total weight and the same weight for the insertion, deletion and substitution penalties).

Once done, the minimum of $e(i, j)$ on the top row gives the best edit distance and query score (the lower, the better). Backtracking from the minimum location gives the location of the instance of the best match.

$$e(i,j) = \min \left\{ \begin{array}{l} e(i-2,j-1) + c_{si}(p_{sub}(d(i-2), q(i-1)) + p_{sub}(d(i), q(i)) + c_{ii} p_{ins}(d(i-1))) \\ e(i-1,j-1) + c_{sn}(p_{sub}(d(i-1), q(i-1)) + p_{sub}(d(i), q(i))) \\ e(i-1,j-2) + c_{sd}(p_{sub}(d(i-1), q(i-2)) + p_{sub}(d(i), q(i)) + c_{dd} p_{del}(q(i-1))) \end{array} \right\} \quad (1)$$

3.3. Variable versus fixed penalties

The phone insertion, deletion and substitution penalties may be either constant or may depend upon the actually inserted, deleted or substituted phones since some phones are more likely than other to be inserted, deleted or substituted. For the fixed penalties, we chose:

- $p_{ins(p_i)} = 1$
- $p_{sub(p_i,p_j)} = 1 - \delta(i,j)$
- $p_{del(p_j)} = 1$

and for the variable penalties, we chose:

- $p_{ins(p_i)} = -\log(\epsilon + prob(insertion(p_i))$
- $p_{sub(p_i,p_j)} = -\log(\epsilon + prob(substitution(p_i,p_j)))$
- $p_{del(p_j)} = -\log(\epsilon + prob(deletion(p_j))$

The probabilities were estimated by comparing manual and automatic transcriptions.

4. Vector Space Model based search

We also made experiments with the Vector Space Model (VSM) classically used in textual information retrieval. We used phone bigrams as the basic indexing unit associated to a pivoted cosine normalization scheme. This normalization was proposed by Singhal, Buckley and Mitra in 1996 [10]. In the vector space model, we usually define, for the weighting scheme, a normalization factor intended to compensate for the fact that long documents have intrinsically more chances to be relevant for a given query. Such normalization may use the document vector norm or the size of the document in terms of characters for instance. In [10], the authors found out that using usual cosine normalization tends to over boost the short documents and to penalize too much the long documents. This is why they proposed a pivoted normalization to compensate for the usual normalization. In our case, documents vary a lot in length, explaining why such pivoted length normalization is needed. Compared to usual normalization slope for text (between 0.2 and 0.3 according to [10]) the slopes of the normalization for our best results using the vector space model are of 0.4 for AT1 and 0.54 for AT2. This normalization slope is quite large; this is probably due to the fact that, on this collection, longer documents need to be boosted during the indexing to be retrieved.

5. Experimentations

5.1. Monolingual search, validation on the Star Challenge development set

The goal of the first series of experiments was to evaluate the relative performance of word-based and phone-based search as well as the benefit brought by the use of variable penalties in the latter case. Three dynamic programming (DP) based methods were compared to some baselines and to a Vector Space Model based methods.

These experiments were carried out on the audio development set of the Star Challenge. This set consists in about two hours of monolingual (English) audio data (233 segments) and 39 solved queries for both AT1 (queries as IPA strings) and AT2 (spoken queries). The systems are asked to return a list on 50 hits and the evaluation metrics is the MAP defined in the Star Challenge for audio search (Eq. 2; this MAP is different from the standard TREC MAP metrics):

$$MAP = \frac{1}{L} \sum_{i=1}^{L} \left(\frac{1}{R_i} \sum_{j=1}^{R_i} \delta(i,j) \right) \quad (2)$$

where L denotes the total number of queries, R_i is the total number of documents relevant to the ith query, and $\delta(i,j)$ is an indicator function which is 1 when there is a hit (i.e. the jth relevant document is in the list output by the retrieval method for query i) and 0 otherwise.

Documents (audio segments) and AT2 queries were transcribed in IPA in two ways. The first one is a transcription at the word level followed by a conversion at the phone level. The second one is a direct transcription at the phone level. After that and in both cases, all documents and queries are constituted of IPA strings.

Several baselines were used for comparison. A random choice is natural and it constituted the baseline 2. Another possibility is to sort the segments according to their length: the longer the segment, the higher its probability to contain the query, regardless of its contents. Baseline 1 corresponds to the choice of the shortest segments (worst case) and baseline 3 corresponds to the choice of the longest segments (best case). All of these baselines actually ignore the contents of the segments and the results are the same for AT1 and AT2 since no use is made of the query contents either. Baseline 4 consists in the search for an exact match of the query string within the segment string. This corresponds to the Unix

"grep" command and also to a dynamic programming based search with infinite deletion, suppression and substitution penalties. Since exact matches are very infrequent and do not provide enough segments for completely filling the result list, additional segments are chosen among the longest ones.

Dynamic programming was tried at the word level with fixed and variable penalties and at the phone level with variable penalties.

Finally, the Vector Space Model (VSM) commonly used in information retrieval was tried. The indexing terms were bigrams of phones with the pivoted cosine normalization weighting scheme.

Method	AT1	AT2
Baseline 1: shortest segments	0.024	0.024
Baseline 2: random	0.242	0.242
Baseline 3: longest segments	0.497	0.497
Baseline 4: "grep" + longest segments	0.557	0.560
DP, word recog., fixed penalties	0.776	0.632
DP, word recog., variable penalties	0.843	0.636
DP, phone recog., variable penalties	0.706	0.650
VSM, word recog., bigrams	0.797	0.660

Table 2. Validation on the Star Challenge development set

Table 2 shows the results obtained for the tested methods and baselines. The following observation can be made:

- the baseline performance increased as we predicted: shortest $<$ random $<$ longest $<$ grep+longest;
- the variable penalties significantly improved the performance;
- as expected too, the phone level transcription produces less good results because it does not benefit from the word-level language model; however, it is the only one that can be available for documents in unknown languages and it is the one that will be used for the multilingual search;
- finally, the VSM based systems performed quite well at the word level but it did less well than baseline 4 at the phone level (not shown).

5.2. Monolingual search, evaluation on the Star Challenge "round 1" data set

The system that performed better on the development set was used for the official submission of the round 1 of the Star Challenge. This is a system using dynamic programming for the search with variable penalties. An additional improvement was made; it consisted in using three transcriptions with different phone bigram weights and averaging over the corresponding DP scores. This leaded to another small improvement in the system performance.

Table 3 shows the results obtained with this system on the round 1 set. This set is composed of 25 hours of monolingual

Data set	Penalties	AT1	AT2	Mean
Devel.	fixed	0.760	0.679	0.719
Devel.	variable	0.858	0.728	0.793
Round 1	fixed	0.643	0.319	0.481
Round 1	variable	0.634	0.324	0.479

Table 3. Influence of the use of variable penalties

(English) audio data (4300) segments and 10 queries for both AT1 and AT2 tasks. For comparisons, results are also shown with the same system with fixed penalties on round 1 data and with fixed and variable penalties on development data. The following observation can be made:

- the performance on round 1 data is much lower than on development data;
- the significant performance gain obtained by the use of variable penalties on development data is not obtained again on round 1 data;
- the performance drop between AT1 and AT2 is much more important on round 1 data.

All these effects are probably due to the fact that round 1 data contains much more audio segments of which a much smaller proportion is relevant, making the task harder. These segments are also smaller. Using this approach, the LIG team ranked first on AT1, third on AT2 and third in total among 35 participating teams.

5.3. Multilingual search, validation on the Star Challenge "round 1" data set

The goal of this series of experiments is to validate the multilingual search using the available training data. Since the target languages were not known we could only validate the approach on the monolingual (English) available data. We did however build models using other languages and tested it on English data, considering that this would be representative enough. We first tried models built from single languages and use them for indexing and retrieval: English (EN), Chinese (CH) and Malay (MY). We also tried a model which is a combination of these three languages and Vietnamese (ML4). We finally tried the same with phone bigrams (BG) and a combination of three models with different language model weighting (Fuse).

	EN	CH	MY	ML4	BG	Fuse
AT1	0.668	0.476	0.428	0.603	0.615	0.650
AT2	0.585	0.578	0.577	0.568	0.591	0.638

Table 4. Results obtained with different language models:

Table 4 shows the results obtained with these different models. The following observation can be made:

- we used a new English model that turned out to be better than the one used for the official submission on round 1, especially for AT2;

- the results obtained with language models that are different from the target language (Chinese or Malay against English) are quite good despite the very different phonetic contents;
- results are better for AT2 than for AT1 in this case; this is probably due to the fact that similar confusions are made during the document and query transcription and that they compensate each other;
- the ML4 multilingual model is almost as good as the purely English model and BG and Fuse do even better though from the way they are built, these models are expected to be as good for Asian languages.

The LIG team used the "Fuse" model for its official submission for the round 3 and ranked fifth on the audio task and first on the multimodal task (combination of IPA based search and image content-based search of video segments), qualifying for the challenge final in Singapore.

6. Conclusion

We have presented an approach based on the use of the International Phonetic Alphabet (IPA) for content-based indexing and retrieval of multilingual audiovisual documents. The approach works even if the languages of the document are unknown. It has been validated in the context of the "Star Challenge" search engine competition organized by the Agency for Science, Technology and Research (A*STAR) of Singapore. Our approach includes the building of an IPA-based multilingual acoustic model and a dynamic programming based method for searching document segments by "IPA string spotting". Dynamic programming allows for retrieving the query string in the document string even with a significant transcription error rate at the phone level. The methods that we developed ranked us as first and third on the monolingual (English) search task, as fifth on the multilingual search task and as first on the multimodal (audio and image) search task.

The results obtained were quite good and validated the principle of the use of IPA transcriptions for the indexing and retrieval of audiovisual documents in unknown languages. Some additional experiments should be done on larger corpora for confirming of the obtained results. Several improvements are still possible both at the level of the building of the multilingual acoustic models and at the level of the dynamic programming based search. On the latter point, a direct search in the phone lattice produced by the transcribing tools appears very promising.

References

[1] D. Moraru, L. Besacier, S. Meignier, C. Fredouille, and J.-F. Bonastre, "Speaker Diarization in the ELISA consortium over the last 4 years," in *RT2004 Fall Workshop*, 13-14 Nov. 2004.

[2] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the cmu-cambridge toolkit," in *Eurospeech'07*, 1997, pp. 2707–2710.

[3] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Intl. Conf. on Spoken Language Processing*, 2002. [Online]. Available: citeseer.ist.psu.edu/stolcke02srilm.html

[4] LDC, "http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?-catalogId=LDC98S71," 1997.

[5] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and Thayer, "The 1996 hub-4 sphinx-3 system," in *In DARPA Speech Recognition Workshop*, Chantilly, VA, February 1997.

[6] LDC, "http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?-catalogId=LDC93S6B," 1993.

[7] CCC, "http://www.dear.com/CCC/resources.htm," 2005.

[8] V.-B. Le, T. Do-Dat, E. Casteli, L. Besacier, and J.-F. Serignat, "Spoken and written language resources for vietnamese," in *LREC'04*, 2004, pp. 599–602.

[9] J.-L. Gauvain and J.-J. Mariani, "A method for connected word recognition and word spotting on a microprocessor," in *Proc. IEEE ICASSP 82*, vol. 2, 3-5 May 1982, pp. 891–894.

[10] A. Singhal, C. Buckley, and A. Mitra, "Pivoted document length normalization," in *ACM SIGIR conference*. ACM Press, 1996, pp. 21–29.