EUROSPEECH '97
5th European Conference on Speech
Communication and Technology
Rhodes, Greece, September 22-25, 1997

AUTOMATIC GENERATION OF HYPERLINKS BETWEEN AUDIO AND TRANSCRIPT

J. Robert-Ribes [1,2,3] *and R.G. Mukhtar* [1,3]
[1] Advanced Computational Systems CRC
[2] Computer Science Lab, Australian National Univ.
[3] C.S.I.R.O. Mathematical and Information Sciences
Locked Bag 17, North Ryde NSW 2113, Australia
FAX: +61 2 9325.3200, E-mail: Jordi.Robert-Ribes@cmis.csiro.au

ABSTRACT

We present a prototype that enables the generation of hyperlinks between audio and the corresponding transcript. The main issue in generating such hyperlinks is determining common time points in the transcript and the audio (this is also called aligning).

The system is speaker independent and can deal with inexact transcripts. It combines inaccurate modules in such a way that the final results are extremely satisfactory.

1. INTRODUCTION

FRANK (Film Researchers Archival Navigation Kit) [6] is a tool created as part of the development of a distributed system for the searching and browsing of digital video collections over wide-area broadband networks. FRANK allows the viewing of the "continuous media" (e.g. the video) and alternate representations that are available (e.g. the transcript or textual version of the speech in the audio track).

To give effective use of the alternate representation (the transcript), we must provide tight coupling of the navigation through the video and the transcript such that the users feel they are navigating through a common information space.

The key attribute linking video and transcripts is time. Therefore in order to have links between a segment of transcript and the corresponding video, we need the time at which the speech of that segment starts.

While manually transcribing a video, it is easy to insert the time at the beginning of each paragraph. But unfortunately, this is not always done and was almost never done in the past.

If automatic continuous speech recognition (speaker independent) was a completely solved problem, it could be used to link the audio document with its transcript. Unfortunately this is not the case.

We have developed a tool for generating such hyperlinks off-line. This tool (in prototype version) is presented in the present paper. It determines anchor points (or time labels) between the transcript and the audio track of the video. These anchors are converted into hyperlinks by inserting the corresponding HTML tags.

Such a tool will be part of the supporting tools that FRANK offers to populate digital film collections. Other supporting tools deal with the video part (e.g. automatic detection of scene changes).

An increasing number of speech researchers have taken interest in accessing large collection of speech data. However, most of the works relate to keyword spotting or indexing (e.g. [1],[9]). Some research has been done on aligning or matching speech and text for large corpus. For instance [4] uses speaker identification to align audio with text transcripts.

In the next section we present the framework for Automatic Linking of Transcript and Audio. Section 3 describes the prototype we have implemented. Section 4 gives the results on the accuracy. The conclusion is given in Section 5.

2. ALTA (AUTOMATIC LINKING OF TRANSCRIPT AND AUDIO)

2.1 Overview

In a general case, we may want to determine anchor points between the audio and a textual description of that audio. The description can indicate, for instance, the types of speech, noises or music. Then an analysis of the audio will detect the starting times of the corresponding speech, noises or music.

In this paper we consider a subproblem of the previous one: linking the audio to the transcription of its speech message. We have named this "Automatic Linking of Transcript and Audio" (ALTA). Even if the transcript deals only with the speech message, the audio has speech as well as non-speech sounds. The automatic linking of audio (with only speech sounds) and its exact transcript is a different problem ([7],[10]).

We have come across big video collections (for instance, TV archives) that have been transcribed. Consequently they have access to the digital audio and the transcripts in electronic form. However, there is often no anchor points between the text and the audio. Even nowadays, a lot of transcriptions do not have any time annotation. ALTA systems will be useful for such collections.

2.2 Types of ALTA

In some cases, the need is to determine the position in the text for a given time in the audio track. This happens, as we observed in a real case, when a transcript of a TV program has to be cut into camera takes for which the starting times are known.

In other cases, the user needs the time in the audio for a given position in the text. This happens for instance when a user is browsing a collection (e.g. oral history) by visually scanning the transcripts and wants to hear a specific segment. For clarity reasons, we will only consider this case.

Both types can need hyperlinks at different rates. For instance, we may need one for every single word, or just one for every paragraph, or even one every minute.

Alternatively, we may only want to insert a hyperlink at the beginning of each paragraph but only if it is certain (to a high degree) that the hyperlink points to the correct part of the sound. In the current paper we will consider such a case.

2.3 Requirements for our ALTA prototype

2.3.1 No need of real-time

In our application of ALTA (as in many others), there is no need for real-time execution. This is due to the fact that the hyperlinks can be added off-line when the material is added to the collection.

2.3.2 No limitation of duration

One piece of material (e.g. a TV program) can be of any length. Therefore our ALTA system must be able to add hyperlinks to material of duration ranging up to several hours.

2.3.3 Loose precision

In most of the cases we looked at, there is no need for a very accurate precision for the alignment. An inaccuracy of 1 or 2 seconds is tolerated. On the other hand, we have noticed that users, navigating/browsing audio (or video) collections, prefer to start listening (viewing) at least 1 second before the exact position that matches the text they are looking for.

Other situations need a very precise alignment, for instance the labeling at the phoneme level of databases for speech analysis. This problem (solved by [7] and [10]) is very different in several ways from the problem dealt in this paper. We will not consider such cases in the present paper.

2.3.4 Any speaker, any sound

The audio can have several speakers with many different accents and cultural backgrounds. Since we cannot train a speaker identification system, we cannot use the techniques used by [5].

In addition, there can be music and noises between segments of speech (or even in the background). The non-speech sounds are not usually represented in the transcript.

2.3.5 Inaccurate transcript

The transcripts are not always exact. We may have some words inserted, some deleted. In some cases, we had to deal with whole paragraphs deleted because the transcript that was archived was not the one corresponding to the final version of that TV program.

Even worse than that, we had to deal with old transcripts in paper version with very bad quality. We obtained an electronic version by using an Optical Character Recognition software but of course this version had a lot of mistakes.

2.3.6 Final remarks

The first implementation that comes to mind would use a traditional Automatic Speech Recognition (ASR) system with a constrained language model. This language model would only allow the word sequence that appears in the transcript. This would be a too restrictive solution for our problem. It would not work due to the inaccuracy of the transcript and the existence of non-speech sounds. Moreover, the implementation should have to be adapted to use such long audio files and process them in a reasonable amount of time.

We developed a new algorithm that is implemented with the prototype we present in the next section.

3. THE ALTA PROTOTYPE FOR FRANK

3.1 Overview

This section will present the prototype we have developed to be used as a support tool for FRANK. It is given the audio track of an MPEG video file and its transcript. It then generates a new transcript file with an hyperlink at the beginning of each paragraph. These hyperlinks point to the corresponding instant in the video. It will only add the hyperlinks for which it is confident about their precision. It is normally preferable to have less hyperlinks but have them correct.

The prototype is built from very simple and inaccurate tools. However, the overall architecture leads to extremely satisfactory results.

The architecture consists of 5 main modules: two windowing modules, the language model generation module, the automatic speech recognition module and the analysis and decision module. The overall structure is presented in Figure 1.

3.2 Windowing modules

Inputs: the full transcript/audio and the window start and end points. Output: a segment of text/audio.

A segment of text is extracted from the full transcript and a segment of audio from the audio file. The starting and ending points of the segments are given by the "Analysis & Decision module".

3.3 Language Model Generation module

Input: a text segment (from 3.2). Outputs: a language model and a dictionary.

The Language Model Generation module will produce (based on the segment of text presented at its input) a language model and a dictionary. It first converts the text into phonemes. The text-to-phoneme converter used is

the one developed by Wasser [8] which is freely available. The phoneme set it uses is not the same as the one used by the ASR module (ANDOSL set [3]). So we used a rough translation table to translate the sequence output of the text-to-phoneme converter into a sequence of phonemes from the ANDOSL set. This sequence of phones is then used to calculate the language model and create a dictionary (each word with one pronunciation).
This module is very inaccurate. Firstly, the text-to-phoneme converter does not take into account Australian accent. Secondly, it only allows one pronunciation per word. And thirdly, the use of a rough translation table adds further inaccuracy.

3.4 ASR (Automatic Speech Recognition) module

Inputs: the audio segment (from 3.2), the language model and the dictionary (from 3.3). Output: a list of words (or filler models) with the corresponding times, acoustic likelihoods and number of phonemes.

We used the HTK 2.5 ([11]) to implement the core of this module based on Hidden Markov Models (HMM). In addition, we added the number of phonemes (taken from the dictionary) to the standard output of the HTK recognition tool.

We use 47 models: one for each of the 44 phonemes used, 1 for pauses, 1 for silences and 1 for non-speech signals.

Word models are constructed from the corresponding monophone models. The filler models are the monophone models (as in [2]).

3.4.1 Topology of the HMMs

Three topologies are used:

- Phonemes. Three states with four mixtures. Left to right with one skip from first emitting state to last emitting state.
- Pause. One emitting state with one mixture. Left to right with one skip allowing the only emitting state to be skipped.
- Silence and non-speech signals. Three emitting states with one mixture, ergodic allowing transition from last to first emitting state and from first to last.

The input parameters used are the energy and 12 Mel Frequency Cepstral Coefficients with the first and second derivatives. This gives a total of 39 values.

3.4.2 Training of the HMMs

The non-speech model was trained on the non-speech sounds contained in only one 45 minutes video. That gave a total of only 2.5 minutes of non-speech sounds used for the training.

The phoneme, pause and silence models were trained with 200 sentences per speaker of 6 male and 6 female young cultured Australian speakers from the ANDOSL corpus ([3]). Total: 2400 training sentences.

The amount of data used for the training is extremely low. Therefore, the models will be far from optimal. The phoneme recognition rate on the training data was around 50% and on some test data (not used for training) was about 20%. When these models are used on the real life data, the correct recognition rate at the phoneme level will be much less.

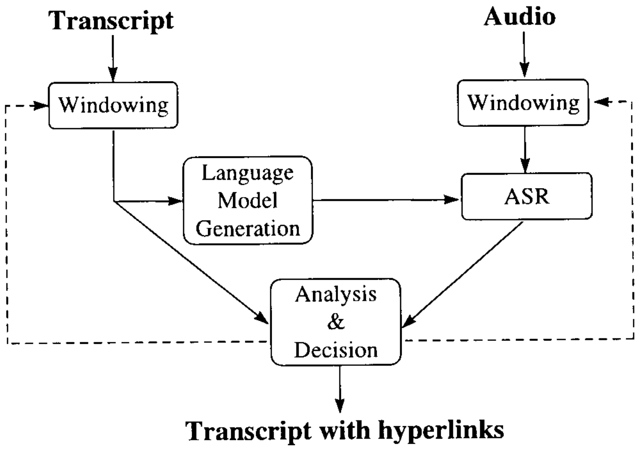

Figure 1 Architecture of the prototype

We can see that the HMMs will introduce a lot of inaccuracy. However, we expect it to be compensated by the whole architecture, mainly the language model (even if inaccurate itself) and the posterior analysis (see 3.5).

3.5 Analysis & Decision module

Inputs: the segment of text (from 3.2) and the list of words, times, likelihoods and number of phonemes (from 3.4). Outputs: the limits for the "Windowing modules" and the original text with the hyperlinks.

The "Analysis & Decision module" is in charge of analysing the output of the ASR module and then deciding if the text segment corresponds to one part of the audio segment. This analysis is done only on the words which have more than a predefined number of phonemes (avoiding the numerous errors in short words). The decision is taken by comparing a threshold with the ratio of the outputs of the word model and the filler models.

If the decision is negative, it updates the limits of the windows. If the maximum number of iterations is exceeded, it considers it cannot locate the corresponding audio. It then outputs the segment of text without hyperlinks. It also updates the windowing of the transcript to the next paragraph and the windowing of the audio to an "approximate guess" of the possible location in the audio track. If the maximum number of iterations is not reached, it extends the audio windowing limits, increments the iteration counter and tries again to locate the corresponding audio.

If the decision is positive, it outputs the paragraph (with the original formatting) and the added hyperlink at the beginning of the paragraph. It then updates the windowing limits of both text and audio.

3.6 Free parameters

The prototype described above has several parameters that can be tuned. The main ones are:

- Window lengths: we may impose minimum and/or maximum lengths to each window in the Windowing modules. If the windows are too short, the corresponding audio may not be inside the windowed signal. If they are too long, the computation time will increase dramatically. In addition, if the audio window is too long, we may locate another occurrence of the same sentence.
- ASR parameters. Since the Hidden Markov Models of the ASR module are implemented with HTK, all the parameters of HTK (see [11]) can be used.
- Minimum number of phonemes. The minimum number of phonemes per word to take into account in the analysis can be set to different values.
- Decision threshold. The decision threshold used in the Analysis & Decision module can also be set.
- Maximum number of iterations. The maximum number of iterations (see 3.5) can have an important role in the processing time. If set too high, impossible searches will be tried many times leading to excessive processing times. If set too low, some sentences will be missed.

4. ACCURACY RESULTS

This section provides some figures on the accuracy of the prototype. The reference taken is the manually inserted hyperlinks that were inserted by an operator in an earlier stage. This manually inserted hyperlinks were roughly created. Therefore they cannot be taken as an absolute reference. We used them because they were the only reference available (apart from manually generating them again very accurately!).

We define a hit as the automatic hyperlink being less than 3 seconds apart from the manual one. The other cases will be considered misses.

The data presented here is taken from two video programs (total duration 1 hour).

The system did not generate hyperlinks for 8.7% of the paragraphs. In fact, the confidence ratio was below the threshold, so no hyperlink was generated. All (100%) of the hyperlinks generated were hits.

We investigated if further inaccuracy could be introduced in the ASR module without losing hyperlink accuracy. We trained another set of HMM with only one male and one female young cultured Australian speakers (total: 400 sentences). In this case, 15.2% of hyperlinks were not generated and 95.7% of the generated ones were hits. Consequently, if the inaccuracy of the ASR module is too high the system will not be able to compensate it.

5. CONCLUSION

We have presented a prototype for automatic generation of hyperlinks between audio data and its speech transcript. Some of the modules are extremely imprecise and inaccurate (Language Model Generation and ASR modules). However, the whole architecture can recover the errors generated by these modules and all the hyperlinks generated be correct.

The architecture can be enhanced by including weighting factors in the decision stage. In the present version the only weighting is binary depending on the number of phonemes in the word. The new weighting could be given by some Natural Language Processing or some statistical analysis of the text. Other improvements can be done to allow multiple pronunciations or better training the Hidden Markov Models. Even if the present accuracy is extraordinary, with better HMMs we should improve the rate of hyperlinks generated. However, the improvement may not be worth the high cost of such training.

REFERENCES

[1] Brown, M.; Foote, J.; Jones, G.; Spärck Jones, K., and Young S., "Open-vocabulary speech indexing for voice and video mail retrieval". ACM Multimedia. 1996: 307-316.

[2] Knill, K. and Young, S., "Fast implementation methods for Viterbi-based word-spotting". ICASSP'96, pp. 522-525, Atlanta, 1996.

[3] Millar, B.; Vonwiller, J.; Harrington, J., and Dermody, P., "The Australian National Database of Spoken Language", ICASSP'94; 1994.

[4] Roy, D. and Malamud, C., "Integration of a large text and audio corpus using speaker identification", AAAI Spring Symposium, Palo Alto. 1997.

[5] Roy, D. and Malamud, C., "Speaker identification based text to audio alignment for an audio retrieval system", ICASSP'97, pp. 1997-1099, Munich, 1997.

[6] Simpson-Young, B. and Yap, K, "FRANK: Trialing a system for remote navigation of film archive", SPIE Symp. Voice Video & Data Communications, Boston. 1996.

[7] Vonwiller, J.; Cleirigh, C.; Garsden, H.; Kumpf, K.; Mountstephens, R., and Rogers, I. "The Development and Application of an Accurate and Flexible Automatic Aligner", Int. J. of Speech Tech. (forthcoming, Vol 1, n. 2).

[8] Wasser, J., "English to phoneme translation", http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/speech/systems/eng2phon/, 1985.

[9] Wechsler, M. and Schäuble, P. "Speech retrieval based on automatic indexing", MIRO'95, 1995.

[10] Wightman, C. and Talkin, D., "The Aligner: Text-to-Speech alignment using Markov models". Eds: van Santen et al. Progress in Speech Synthesis. New-York: Springer-Verlag; pp. 313-323; 1997.

[11] Young, S.; Odell, J.; Ollason, D.; Valtchev, V. and Woodland, P., "The HTK Book v2.1." Entropic Research Laboratories Inc. 1995.

ACKNOWLEDGEMENT

The authors wish to acknowledge that this work was carried out within the Cooperative Research Centre for Advanced Computational Systems established under the Australian Government's Cooperative Research Centre (CRC) Program.

We also acknowledge the help of the staff of the Computer Science Laboratory at the Australian National University.