A FACTOR AUTOMATON APPROACH FOR THE FORCED ALIGNMENT OF LONG SPEECH RECORDINGS

Pedro J. Moreno, Christopher Alberti

Speech Research Group, Google Inc.
76 Ninth Avenue, New York, NY 10011.

ABSTRACT

This paper addresses the problem of aligning long speech recordings to their transcripts. Previous work has focused on using highly tuned language models trained on the transcripts to reduce the search space. In this paper we propose the use of a factor automaton, a well known method to represent all substrings from a string. This automaton encodes a highly constrained language model trained on the transcripts. We show competitive results with n-gram models in several testing scenarios. Preliminary experiments show perfect alignments at a reduced computational load and with a smaller memory footprint when compared to n-gram models.

Index Terms— finite state transducers, speech alignment, speech recognition.

1. INTRODUCTION

In recent years large amounts of audio and video content have become available over the Internet. Podcasts, audio books, video sharing web sites such as YouTube, digital archives of radio and television, university lectures and other web resources have made it easier for users and institutions to share multimedia content. Interestingly much of this content does contain associated transcripts. These resources have turned into an extremely useful source of information to speech researchers. Mining these data sources, either to facilitate their search via audio indexing engines, or to improve the performance of large vocabulary speech recognition systems has become an interesting area of research. In this paper we introduce a new technique to address this data mining problem.

Aligning speech to its transcript is a simple problem with a well known solution in speech recognition systems. Given properly trained acoustic models, a dictionary mapping words to phonemes, and a word transcript, the Viterbi algorithm gives us an accurate solution to this problem. This solution fails however when the audio segment is longer than a few minutes, or when the transcripts are not completely accurate as is the case of closed captions. The length of the audio poses a problem to decoding engines as the memory requirements of most Viterbi search algorithms quickly becomes impractical on long audio recordings. Transcriptions with high error rates can also lead any recognition engine in the wrong direction. To address these limitations, researchers have turned the alignment problem into a large vocabulary speech recognition problem.

First, to address the problem of long audio streams most solutions break the audio stream into smaller segments (see for example [1] or [2]) often looking for silences as potential breaking points. Secondly, to address the problem of potential mistakes in the transcript and to reduce the search space most solutions build n-gram language models. These n-gram language models are either trained only on the transcripts [1] or trained on a combination of the transcripts and background general English language models [2, 3]. In this paper we propose a new approach to aligning speech to audio. Rather than using n-grams we propose the use of factor automata to encode the transcript.

The outline of the paper is as follows. In Section 2 we give a brief description of factor automata and how they can be applied to aligning speech to text. In Section 3 we describe our experimental setup with a description of the databases used. In Section 4 we report on our experimental results. We conclude the paper in Section 5 with a summary of results and potential ideas to explore in future research.

2. FACTOR AUTOMATA

Finite state automata are widely used in many fields, e.g. speech recognition, text processing, and computational biology. In speech recognition they are often used to represent dictionaries, language models, and the mapping of triphones to phonetic sequences. An automaton accepting the factors, or substrings, of a set of strings is known as a factor automaton.

Factor automata have been used before as efficient indexing structures in many problems such as audio indexing [4] and music search [5]. They are a compact representation of the indexed corpus and since the lookup time in a factor automaton is linear in the size of the query, they enable optimal retrieval performance for indexing tasks [6]. Their con-

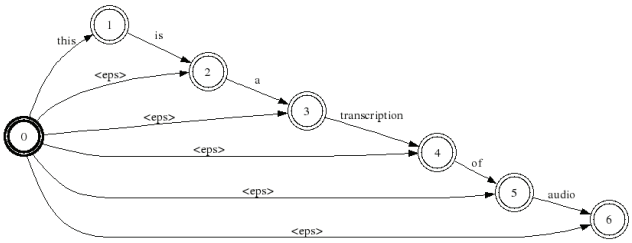

Fig. 1. A factor automaton for a small text example. The initial state is 0. Double circles denote final states.

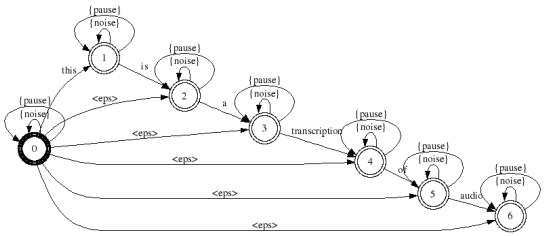

Fig. 2. A factor automaton allowing insertions, modeled as noise.

struction, scalability and theoretical properties are well studied (see e.g. [7]).

In the following discussion we will focus on the specific case of the construction of a factor automaton for a single string. Let $\Delta = \{word_1, word_2, \ldots, word_m\}$ denote the vocabulary of our recognizer, and let $x \in \Delta^*$ be the sequence of words in the transcript to which we want to align the audio. A *factor* or *substring* of x in the alphabet Δ is a sequence of consecutive words appearing in x. Thus, y is a factor of x iff there exist $u, v \in \Delta^*$ such that $x = uyv$. Let $F(x)$ be the deterministic and minimal factor automaton of x, i.e. one that accepts all the factors of x and no other strings. If x is a string of N words, $F(x)$ contains at most $2N - 2$ states and $3N-4$ transitions [8, 9]. Figure 1 shows the factor automaton constructed on the sentence "this is a transcription of audio".

Our factor automaton can be efficiently compacted and determinized, limiting the number of possible transitions at any state to the vocabulary size. This restricts the search space of alternative hypotheses considered by the recognizer at runtime, resulting in an efficient alignment algorithm. In addition since all substrings of the true transcription are allowed, the algorithm is well matched to the use of a segmenter (see Section 3). More specifically, the decoding process implicitly aligns the transcription of a segment of a long audio file to its precise location in the reference transcription. Thus a factor automaton can be viewed as an unweighted$n$-gram language model accepting only those n-grams seen in the reference transcript (i.e., no backoff), from length N (i.e., all words) to length 1. A regular n-gram language model by its nature overgenerates, as it accepts transcriptions that are not contained in the reference transcript.

The factor automaton is robust to word deletions, i.e., words transcribed but not spoken, and substitutions because of the limited number of hypotheses it allows, as long as the transcript is reasonably close to the recording. With minimal modifications it can also model word insertions, i.e., words spoken and not transcribed, and account for them by allowing a noise between any two words. Figure 2 shows a modified version of the automaton where a self loop at each state with accepting symbol $\{noise\}$ can model unknown words or background noises.

3. SYSTEM DESCRIPTION

We use the Google speech indexing infrastructure to perform all our experiments. We start by segmenting the audio into smaller units, looking for silences or other potential break points. A classifier also rejects segments that are considered too noisy, or that contain too much music. We follow a similar approach to the one described in [10].

The audio chunks that are not rejected by the segmenter are then sent to Google's large vocabulary speech recognition system. The Google speech recognition engine uses standard PLP features. These are LDA rotated and their statistics modeled using GMM-based triphone HMMs, decision trees, STC and an FST-based search. Transducers are used to represent the language models, dictionary, and triphone to phone mappings. They are combined in a single static transducer network.

For audio alignment our recognizer also uses two specially designed modules. The first one is a pronunciation module that generates dictionary entries for out of vocabulary words found in the transcript. We use a variant of the pronunciation by analogy algorithm (see for example [11]). The second module builds a language model based on the transcript. It can build a factor automaton with additional silence or noise self loops at each state, an n-gram language model based on the transcripts alone, or an n-gram language model by interpolating the transcripts and an existing general English language model.

The segmenter and recognizer are encapsulated in a replicated server architecture. A client sends complete audio recordings with their transcripts to these servers and collects the global transcript for each show. Error rates are measured comparing the ground truth with the returned hypotheses. Figure 3 gives a high level view of the alignment architecture.

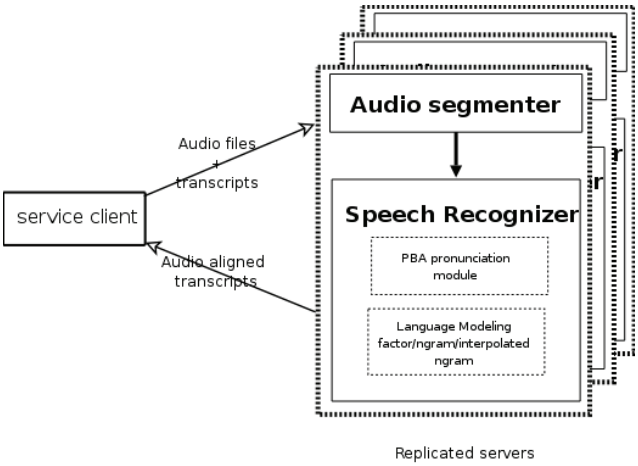

Fig. 3. System architecture for speech to text alignment.

4. EXPERIMENTAL RESULTS

To assess the quality of our new approach we experimented with a variety of recordings. We experimented first with a single recording of English broadcast extracted from public YouTube archives and professionally transcribed. Our aim with this first experiment was mostly to validate our technique. We experimented first with a regular n-gram trained on the existing transcripts to compare with our factor automaton approach. We started comparing n-grams of different lengths with no discounting or cutoffs. We tried n-gram language models going as high as 6-grams and as low as 4-grams. As Table 1 shows there was almost no effect in word error rate due to different n-gram model sizes. The factor automaton had a reduced error rate when compared with the n-gram models.

As we expected intuitively, the errors made by the factor automata were due to deletions and insertions at the border of the utterances, where the constraint represented by the factor automaton is weakest. Part of the errors are also caused by the segmenter labeling as noise or music portions of the audio for which reference transcriptions were given.

LM approach	n-gram order	WER
N-GRAM	6	5.4%
N-GRAM	5	5.4%
N-GRAM	4	5.4%
FACTOR	N/A	2.5%

Table 1. Comparison of WER on a single 10 minute broadcast news video.

Word error rates are computed by measuring the edit distance between the transcripts (on which the factor automaton and the n-gram models are trained) and the hypothesized alignment returned by the recognizer. Word error rate is not the natural way of quantifying the accuracy of a speech alignment system. In this type of experiment however where the transcript is trusted, we found that the timing information produced by the system is usually accurate whenever there is a match in transcripts and hypotheses. Thus the WER is an approximation of the number of words correctly aligned.

As we expected, the factor automata limited the search space significantly. We experimented with replacing the acoustic model with a much simpler acoustic model, where only context independent models with 32 Gaussians per state where used. This resulted in a 150 state model. Even when used with such a simple acoustic model, the factor transducer produced almost identical results with a word error rate of 2.6% (as opposed to the 2.5% WER reported in the bottom row of Table 1). Also, when comparing the CPU time it took to recognize the 10 minute video using n-gram models and a factor model, we measured a 65% reduction with the factor automaton. The reduction in the size of the model is also significant. In our system, the size of an automaton for large vocabulary automatic speech recognition is of the order of 1GB. The specialized n-gram model for the alignment of the broadcast under consideration gave a 7.9MB decoding graph. Using a factor graph and a context independent acoustic model we lowered the size the of the graph to just 577KB.

In a second series of experiments we expanded the size of the test set to 50 different video recordings. These corresponded to lectures of varied length (from 30 minutes to 2 hours) and acoustic conditions recorded live at Google and available via YouTube. Some of these lectures contain highly degraded speech due to poor microphones, or competing speakers in the background, or highly accented foreign speakers. In other cases the speech saturated the recording microphone producing highly distorted speech. In general the recordings were of average to bad quality. The lectures were professionally transcribed although we still found some mistakes in the transcriptions.

Table 2 shows a comparison of results obtained with several n-gram models and a factor automaton using the context dependent language model. In this Table we measure the percentage of video shows with a word error rate below 10%, between 10% and 20% and above 20%. Looking at the results we can see that any n-gram model of order 4 or higher produces identical results. Only the 2gram model yields lower performance. The performance of n-grams and the factor automaton is quite comparable. However the factor automaton is significantly faster. This is due to the reduced size of the automaton and its lower fan-out.

Further analysis of the error showed that most are due to the segmenter rejecting noisy or speech under music, low

LM approach	% videos WER $< 10\%$	% videos WER $\in [10, 20]\%$	% videos WER $> 20\%$
2-GRAM	0	38	62
4-GRAM	20	60	20
6-GRAM	20	60	20
8-GRAM	20	60	20
FACTOR	14	68	18

Table 2. Word error rate for 50 different audio recordings. Increasing the order of the n-gram model beyond 4 did not improve the alignment. Switching to a factor automaton model gave comparable results, but with greatly reduced computational cost.

quality speech (strong noise), and accented speech. In fact these causes of error are common to all alignment techniques considered in this paper.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced a new approach to align speech to its transcripts. We have shown similar performance to previous n-gram based approaches. Our approach has the added advantage of producing smaller and more constrained (lower perplexity) language models resulting in faster processing.

We have observed that in some cases with specially noisy speech segments both n-gram and factor automata language models fail. A simple recursive approach can handle these uncertainties. The approach works as follows: in a first pass the reference transcript and the recognizer hypothesis are aligned and regions that match are identified. Regions between these islands of confidence are reprocessed with more restrictive factor automata (or n-gram language models). The process is iterated until all audio is aligned. See [1] for more details.

We expect this new speech alignment system to become a useful tool in Google's speech indexing effort. In particular the promising results we have obtained when using a factor automaton in combination with context independent acoustic models suggest a possible use in preparing acoustic training corpora in foreign languages. Among many possibilities we also plan to explore its use in validating the quality of transcripts produced by human transcribers. Finally in our audio indexing work we plan to use this technique in any situation in which transcripts are available.

6. ACKNOWLEDGMENTS

We thank the members of the speech team for their help and support during this work. In particular we thank Michiel Bacchiani, Ciprian Chelba, Masha Shugrina, Hank Liao, Olivier Siohan and Eugene Weinstein for useful discussions. We also thank Johan Schalkwyk, Mike Riley and Cyril Allauzen for their help on finite state transducers supported via the OpenFST library [12].

7. REFERENCES

[1] P. Moreno, C. Joerg, J. M. Thong, and O. Glickman, "A recursive algorithm for the forced alignment of very long audio segments," in *Proceedings ICSLP*, 1998.

[2] L. Lamel, J. L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech and Language*, vol. 16, pp. 115–129, 2002.

[3] T. J. Hazen, "Automatic alignment and error correction of human generated transcripts for long speech recordings," in *Proceedings INTERSPEECH*, 2006.

[4] C. Allauzen, M. Mohri, and M. Saraclar, "General indexation of weighted automata: Application to spoken utterance retrieval," in *Proceedings of the workshop on interdisciplinary approaches to speech indexing and retrieval HLT/NAACL*, 2004.

[5] E. Weinstein and P. Moreno, "Music identification with weighted finite-state transducers," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007.

[6] M. Crochemore and W. Rytter, *Jewels of stringology*, World Scientific Publishing Co. Inc., River Edge, NJ, 2003.

[7] M. Mohri, P. Moreno, and E. Weinstein, "Factor automata of automata and applications," in *Proceedeing of the International Conference on Implementation and Application of Automata (CIAA)*, 2007.

[8] M. Crochemore, "Transducers and repetitions," *Theoretical Computer Science*, vol. 45, pp. 63–86, 1986.

[9] A. Blumer, J. Blumer, D. Haussler, A. Ehrenfeucht, M. T. Chen, and J. Seiferas, "The smallest automaton recognizing the subwords of a text," *Theoretical Computer Science*, vol. 40, pp. 31–55, 1985.

[10] X. Zhu, C. Barras, S. Meignier, and J. L. Gauvain, "Combining speaker identification and bic for speaker diarization," in *Proceedings INTERSPEECH*, 2005.

[11] R. Sproat, "Corpus-based methods and hand-build methods," in *Proceedings ICSLP*, 2000.

[12] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFST: A general and efficient weighted finite-state transducer library," in *CIAA*, 2007, pp. 11–23.