Proceedings of ICSLP'96

Cheating with Imperfect Transcripts

Paul Placeway and John Lafferty

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

Most speech recognition systems try to reconstruct a word sequence given an acoustic input, using prior information about the language being spoken. In some cases, there is more information available to the decoder than simply the acoustics. When decoding a television news broadcast, for example, the closed-caption information that is often recorded for hearing impaired viewers may also be available. While these captions are generally not completely accurate transcriptions, they can be considered to be a strong hint as to what was actually spoken.

In this paper, we present a formalization of this problem in terms of the source channel paradigm. We propose a simple translation model for mapping caption sequences to word sequences which updates the language model with the prior information inherent in the captions. We also describe an efficient implementation of the search in a Viterbi decoder, and present results using this system in the broadcast news domain.

1. Introduction

The Informedia News on Demand project [5, 6] is concerned with storing and indexing television or radio news broadcasts, so that they may be searched and browsed in a convenient way. As part of the indexing, the Sphinx speech recognition system [8, 4] is used to produce a text transcript of the audio. When processing television news broadcasts, we may be fortunate enough to receive closed-caption text along with the video and audio tracks. The closed-captions are not a true word-for-word transcript of the audio, but they are often close. The work reported in this paper was motivated by the desire to use these captions, when they are available, to improve the recognition accuracy of the decoder.

This research was sponsored by the Department of the Navy, Naval Research Laboratory under Grant No. N00014-93-1-2005, and the National Science Foundation and the Advanced Research Projects Agency under Grant No. IRI-9314969. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

The approach that we describe treats the acoustic and closed-caption information as the output of a noisy channel. A news transcript E is transformed by the anchor person into an acoustic realization A, possibly together with some visual information. When a human annotator creates a closed-caption transcription C of the broadcast, there are several factors that may affect the words that are produced. The annotator may take visual clues from the broadcast, or be affected by background noise. There could be time constraints or other factors that result in words being omitted, judged to be unimportant, or annotated incorrectly. Different annotators will have varying styles of annotation. We treat the relationship between the acoustics, the actual word sequence of the broadcast, and the closed-caption annotation in statistical terms, modeled as a conditional probability distribution $P(C, A \mid E)$. The assumptions that allow us to efficiently incorporate the closed-caption information into an acoustic decoder which inverts this noisy channel are discussed in the following section.

The scenario where there is information in addition to the acoustic signal that appears in the most basic speech recognition problem may be quite common. Indeed, the framework for closed-captions described above is directly analogous to the use of a translation model to aid a speech decoder for machine-aided translation, as proposed in [2, 3]. Just as a translator's workstation might incorporate a statistical translation model together with an acoustic model so that the translator can dictate his translations, we can imagine that a transcriber's workstation would be equipped with a statistical closed-caption model, to enable highly accurate and fast dictation of news broadcast transcriptions. The aligned captions might also be used for partially-supervised adaptation, to enable the acoustic models to be bootstrapped to a series of increasingly accurate models.

In the following section we describe the noisy channel model for incorporating closed-caption information in general, and the simple finite-state model that we actually used. Section 3 presents the details of incorporating the closed-caption model into the decoder. Section 4 presents the results of experiments carried out using this approach to decode actual news broadcasts.

2. Statistical Cheating

When recognizing speech for which closed-captions are available, the decoder can make use not only of the output A of the acoustic channel, but also the "hint" H that the caption provides, viewed as the output of an imperfect acoustic decoder for the channel:

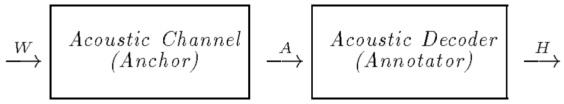

Figure 1: Channel model for broadcasts

The acoustic decoder, in this case, may be a human annotator, listening to the broadcast and providing a rough transcription, subject to time constraints or other conditions that compromise the quality or detail of the annotation. This output, together with the acoustics, can be used by a second decoder, in this case our computer, to decipher the acoustics using the hint provided by the human annotator.

To determine the most probable input sequence W, we apply maximum likelihood decoding in the form

$$\begin{aligned}\hat{W} &= \operatorname*{argmax}_W P(W \mid A, H) \\ &= \operatorname*{argmax}_W P(A \mid H, W)\, P(W)\, P(H \mid W)\,.\end{aligned}$$

The prior probabilities $P(W)$ are given by the source language model, as usual, but the term $P(H \mid W)$ is viewed as a *translation model* of the composite channel which maps source strings W to hints H. While the knowledge of both H and the hypothesized W may be useful for the acoustic model, it is reasonable and convenient to assume that A and H are independent given W, so that the original acoustic models can be used unchanged. The task of the acoustic decoder is then to search for the word sequence W which maximizes the product of the language model, translation model, and acoustic model probabilities:

$$\hat{W} = \operatorname*{argmax}_W P(A \mid W)\, P(H \mid W)\, P(W)\,.$$

Note that by ignoring the hint H through the use of a translation model for which H and W are independent, we arrive at the standard speech decoding problem.

The use of an explicit model for generating captions from text is an important ingredient in our approach. A simple alternative would be to interpolate the text of the closed-captions into a large language model [7]. We tried this, but found that when the new text was given a small interpolation weight, the resulting language model did not improve recognition much, and when the new text was given a large weight, the decoder would often "wander off track," hypothesizing incorrect word sequences from which it had difficulty recovering. An explicit model of how transcripts are converted into

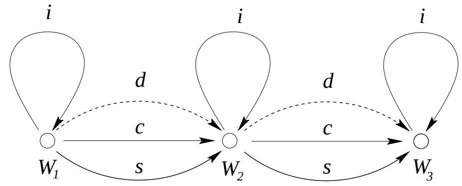

Figure 2: Finite-state translation model

closed-captions provides soft constraits which allow more efficient and accurate decoding of the acoustic signal.

For the actual decoder that we have experimented with, we adopted a simple Markov chain for the translation model, with distinct arcs for matching, mismatching, deleting, and inserting hypothesis words against caption words.

The arc labeled c corresponds to correctly matching a hypothesized caption word C against a particular word in the transcript, W_i. The s arc is taken when a substitution takes place between caption word C_j and transcript word W_i, and a d arc corresponds to deleting (*e.g.* skipping over) word C_j from the captions. Finally, the i arc is taken when there is an insertion of a caption word that does not appear in the transcript. This simple model corresponds to the string edit distance dynamic program, with words as units.

This hidden Markov model for translating transcripts into closed-captions can be combined with the trigram language model to yield a conditional distribution $P(w_i \mid w_1 \ldots w_{i-1}\,;\, c_1 \ldots c_m)$ that w_i is the next word in a transcript whose closed-caption rendering is $c_1, \ldots c_m$, and which begins $w_1, \ldots, w_{i-1}$. This distribution takes the place of the language model in the decoder. The calculation is similar to, but simpler than, that described in [3] using a statistical translation model between French and English. The simplicity comes from the fact that the model proceeds from left to right through both strings, so that there are no crossings in the "alignment" between them.

3. Caption model decoding

Several considerations needed to be addressed in order to integrate the caption model into the Sphinx decoder. The caption text is set before the search commences, and can be considered as a whole for each utterance. The decoder carries out a beam search, so an implementation of this model must be able to match hypothesized words against the closed-caption text incrementally, with multiple partial search paths active at any given time. So the entire state of any caption model must be encapsulated, and this state must be relatively small, for otherwise storing information about every possible alignment would require too much space to be practical.

One option for meeting these criteria is a stack search – a limited-size priority queue of the most likely CC alignments. This proved to be relatively difficult to implement, and quite inefficient due to the burden of maintaining the priority queue. The second was to store a bracketing of alignment scores surrounding the best score at any moment. Suppose we were directly solving the above dynamic program (in the obvious way), iterating over the columns, and then over each element of a column. If finding the actual alignment is not a concern, but only the edit distance is required, then we need only keep two rows of storage, and alternate between them considering one to contain the 'current' row scores and the other to contain the 'next' scores. As an approximation, in place of an entire row of states, we keep only a bracketing of the k scores surrounding the best current state. (So if j is the index of the best scores, we keep indices $j - k/2$ through $j + k/2$.)

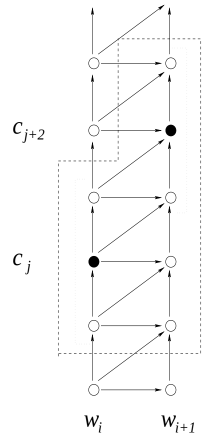

Figure 3: Bracketing of states

In the Sphinx-3 decoder, a theory representing a sequence of words has two components: the current score, and a pointer to a word history record. The word history record forms a linked-list from which a complete word sequence can be recovered. These are created whenever a theory successfully traverses a word to the end without being pruned.

We modified the decoder in two places. Because the word history records are assumed to contain all the information about this word sequence, we added the caption model state to this record. The caption model is evaluated twice for each word: once when starting the word, to start the word with the correct score, and again when that word ends, to correctly record the caption model state. We do not need the whole state of the caption model at word start, only the best score, so we perform an optimized score calculation at this point. This optimization works because we are starting a unique word; other methods are required when using a search vocabulary represented as a tree [10, 9].

Usually a score is augmented by the best possible partial caption score for the relevant word. (That is, the difference between the best caption-model score after adding this word, and the previous best caption score before adding this word.) The only special case is that when the end-of-utterance pseudo-word is started, the caption model must be evaluated all the way to the end of the captions, so that the caption-model contribution to the total score is the total score for the appropriate word history

All other things being equal, this method slows down the search by about 10%, almost entirely due to work done at word-start time. But the search is much more constrained, so we generally see a modest increase in overall speed.

In-practice Use

There is one remaining issue. The acoustic and closed-caption inputs are not closely time aligned, so the actual input is a half hour or more of digital recording, and a body of text considered to be a very long string.

At the moment, the Sphinx decoder cannot handle arbitrarily long inputs. Fortunately, we can side-step this problem. The general method is to cut the recording into a set of short pieces, using characteristics of the signal, in this case periods of low amplitude, to find good separating points in the input.

To align the caption string to the short pieces, we run the decoder without the new CC model over all of the input pieces. Then we find the global minimum string edit distance alignment between the decoder results on the acoustic pieces and the caption text. This turns out to work remarkably well, and good alignments are obtained even when the recognition gets 70% of the words wrong.

4. Experimental results

To set the CC model parameters, we selected a recording of the NBC Nightly News from April of 1995. We measured the correct, insertion, deletion, and substitution rates for the closed-captions, considered as a word sequence, compared to a hand-generated correct transcription of this broadcast, and set the CC model parameters according to these rates, normalizing the sum to 1. We also found that we improved overall performance by setting the CC model 'i' and 'd' costs to be the average of the above insert and delete probabilities.

Our initial experiment for the joint search used a recording of The CBS Evening News from May of 1995. We used the Sphinx-3 decoder, running semi-continuous models trained on WSJ data, with CDCN noise modeling [1]. Our test recording was compressed with MPEG level-2 audio compression at a relatively high bit-rate. (Other experiments showed that this compression does not significantly effect the acoustic performance of the Sphinx decoder.) Table 1 shows results for the section of this broadcast where an anchor (Either Dan Rather or Connie Chung) was speaking exclusively, a total of 378 seconds of data.

For these first two experiments, we hand-aligned and hand-segmented the closed-captions to the correct transcriptions.

	static LM	*LM interpolated with CC data*
plain recognizer	59.8%	47.8%
with CC model	28.5%	18.2%

Table 1: Results on CBS Anchor speech

	static LM	*LM interpolated with CC data*
plain recognizer	16.8%	5.8%
with CC model	4.2%	2.9%

Table 2: Re-recorded speech

	static LM	*LM interpolated with CC data*
plain recognizer	35.9%	21.7%
with CC model	17.2%	10.4%

Table 3: CNN anchor-only speech

	static LM	*LM interpolated with CC data*
plain recognizer	55.8%	47.2%
with CC model	50.5%	35.0%

Table 4: CNN all captioned segments

(Automatic alignments were used for the experiment below.) The word-error rate for the anchor-only section of these closed-captions is 6.6%.

For a baseline, we selected a both-gendered, speaker independent acoustic model (trained on WSJ0 and WSJ1 data), and two language models built from collected broadcast news transcripts from 1994, one with roughly 55 thousand words, the other with the most common 20 thousand. Most lexical transcriptions were hand written, though a number of words that were novel to the broadcast data were automatically generated from a text-to-speech system.

We tried interpolating our general language model with a small one built from the closed-caption text of this broadcast. This was initially done to compare the caption model against this simpler technique. We discovered that the two methods actually complement each other; our best results were obtained using both the interpolated language model and the caption model, along with some acoustic adaptation.

To factor out differences between the acoustic environment of the broadcast news and the environment which the acoustic model was trained for, Bob Weide was kind enough to read the transcripts for this same broadcast, attempting to preserve the original intonation, but using the same recording environment used for the acoustic training data.

The results in table 2 show that in a favorable acoustic environment, and combining language model interpolation with the caption model, we can get very good performance – in this case, more than a factor of five better than the baseline acoustic word error rate, and more than twice as good as the captions alone.

To show the viability of the complete system, we collected another broadcast, CNN Headline News from August of 1995. The audio was processed as above. The output of a decoder run on this data was used to align the caption text. As before, this caption text was also used to build an interpolated language model. The error rate of the captions was 9.7% for the anchor-only segments, and 19.3% for all portions for which captioning was available (everything except commercials). The results for this experiment are shown in tables 3 and 4.

The aligned captions can also be used to help the acoustic model, by doing partially-supervised adaptation. By doing this adaptation on the acoustic model, and combining this with the interpolated language model and caption model, we achieved a word error rate of 8.2% on the anchor-only segments; a 17% relative improvement over the caption text word error rate.

Acknowledgements

Thanks to Alex Hauptmann and Michael Witbrock who provided an interesting problem and valuable insight. Alex also provided the tools to separate a long digital recording into 15–30 second pieces (trying to avoid cutting words apart.) Michael also suggested the technique to align the closed-caption string to the acoustics. Ravi Mosur wrote the Sphinx-3 decoder and described how its data structures function. Bob Weide transcribed and then re-read one show in its entirety.

5. REFERENCES

1. A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition.* Kluwer Academic Publishers, Boston, 1993.
2. J. Brousseau, G. Foster, P. Isabelle, R. Kuhn, Y. Normandin, and P. Plamondon. French speech recognition in an automatic dictation system for translators: the

TransTalk project. In *Proceedings of Eurospeech*, pages 193–196, Madrid, 1995.

3. P. Brown, S. Chen, S. Della Pietra, V. Della Pietra, S. Kehler, and R. Mercer. Automatic speech recognition in machine aided translation. *Computer Speech and Language*, 8:177–187, 1994.
4. L. Chase, R. Rosenfeld, A. Hauptmann, M. Ravishankar, E. Thayer, P. Placeway, R. Weide, and C. Lu. Improvements in language, lexical, and phonetic modeling in Sphinx-II. In *Proc. Spoken Language Systems Technology Workshop*. Morgan Kaufmann Publishers, 1995.
5. M. Christel, T. Kanade, M. Mauldin, R. Reddy, M. Sirbu, S. Stevens, and H Wactlar. Informedia digital video library. *Comm. ACM*, 38(4):57–58, 1995.
6. A. G. Hauptmann, M. J. Witbrock, A. I. Rudnicky, and S. Reed. Speech for multimedia information retrieval. In *UIST-95 Proceedings of User Interface Software and Technology*, 1995.
7. F. Jelinek. Self-organized language modeling for speech recognition. In *Readings in Speech Recognition*. Morgan Kaufmann Publishers, 1990.
8. K. F. Lee. *Automatic Speech Recognition: The Development of the SPHINX SYSTEM*. Kluwer Academic Publishers, Boston, 1989.
9. Mosur Ravishankar. Efficient algorithms for speech recognition. PhD Thesis CMU-CS-96-138, Carnegie Mellon University, 1996.
10. H. Ney, R. Haeb-Umbach, and B.-H. Tran. Improvements in beam search for 10000-word continuous speech recognition. In *Proc. ICASSP-92*, volume I, pages I–9 – I–12, 1992.