Using Quick Transcriptions to Improve Conversational Speech Models

Owen Kimball¹, Chia-Lin Kao¹, Rukmini Iyer², Teodoro Arvizoo and John Makhoul¹

¹BBN Technologies, ²Nuance Communications

Abstract

Using large amounts of training data may prove to be critical to attaining very low error rates in conversational speech recognition. Recent collection efforts by the LDC[1] have produced a large corpus of such data, but to be useful, it must be transcribed. Historically, the cost of transcribing conversational speech has been very high, leading us to consider quick transcription methods that are significantly faster and less expensive than traditional methods. We describe the conventions used in transcription and an automatic utterance segmentation algorithm that provides necessary timing information. Experiments with models trained on a 20-hour set demonstrate that quick transcription works as well as careful transcription, even though the quick transcripts are produced roughly eight times as fast. We also show that when added to a large corpus of carefully transcribed data, quickly transcribed data gives significant improvements in a state-of-the-art ASR system.

1. Introduction

Increasing the amount of training data in automatic speech recognition (ASR) systems has been shown to be an effective way to improve performance in difficult domains. For conversational telephone speech (CTS) recognition systems, however, there are two hurdles to obtaining more training: collecting the data and transcribing it. The Linguistic Data Consortium (LDC) has recently addressed the first issue, collecting the Fisher corpus, a set of over two thousand hours of two-channel English telephone conversations [1]. Using this data in ASR training, however, requires transcribing it, and transcription of CTS data has historically been very expensive, with typical labor estimates of roughly 50 times real-time by trained linguists. For a set as large as the Fisher corpus, this would be prohibitively expensive.

Automatic methods have been shown to reduce or eliminate transcription costs for some domains, but their applicability to CTS data is questionable. In [2], CTS data was automatically transcribed starting from bootstrap models trained on very limited data. However, the low accuracy of the initial models and the lack of language modeling data led to high error rates and a correspondingly low yield of useable training transcripts. In [3] it was shown that for Broadcast News (BN), it is possible to bootstrap from a small, transcribed corpus to obtain transcripts suitable for training high accuracy models. Similarly, recent work demonstrated good results for using unsupervised transcription to adapt from a quiet to a noisy domain [4], though here the target domain had a fairly constrained language model. The difficulty of replicating the successes of automatic transcription in the CTS domain may be due to the higher perplexity of the domain's language model or the greater informality typical of CTS speech.

In this work we looked at using manual transcriptions, but we re-examined the assumption that CTS ASR systems require very careful, very expensive transcription. We investigated a quick transcription alternative that reduced time and cost, yet maintained very good quality for training data.

One way to reduce transcription cost is to develop software tools that integrate automatic utterance segmentation with rapid human transcription, and the LDC is currently exploring such an approach. In order to produce large volumes of low-cost transcripts, however, we chose to work with commercial transcription services. The high volume and low cost of these services allowed us to transcribe the large Fisher training set quickly, but required that we provide the time segmentation of the data.

Specifically, the commercial services we encountered used only simple word processing software and their transcribers were not trained to record utterance endpoint times. Adopting new software that would support easier recording of time information was viewed by the commercial services as impractically expensive, primarily in terms of transcriber retraining cost. Consequently, although using better tools at first seemed like an obvious approach, practical cost considerations led us to split the work into pure transcription without time information, followed by an automatic segmentation and time alignment of the transcripts to audio. The question we tried to answer in this work is whether the resulting quick transcriptions and segmentation would hurt ASR performance in comparison with the more typical careful transcription.

The rest of this paper is organized as follows. In Section 2, we discuss the conventions we asked the transcribers to use in quickly transcribing the data. Section 3 gives details of the automatic algorithm used to segment the transcripts. Section 4 gives results of experiments comparing quick versus careful transcripts. Section 5 illustrates how the resulting quick transcriptions can improve the performance of a state-of-the-art ASR system, and Section 6 gives some conclusions.

2. Quick transcription conventions

In formulating guidelines for quick transcription of CTS data, we started from the instructions Mississippi State University (MSU) used [5] in their work with the Switchboard corpus [6] below. We tried to eliminate instructions that took time without adding information critical for training acoustic models. Transcribers were instructed to label each utterance of the conversation with either “L” or “R”, depending on whether the speech played out in the left or right channel of their headphones, and to put each utterance on a single line of the
transcript. The issue of where best to break speaker turns when time information is not given is somewhat ambiguous, especially in overlapped speech. Our instructions gave some general guidelines, but told transcribers not to worry excessively about the break points since our automatic algorithm, described below, is largely insensitive to their exact location.

As with previous guidelines, we instructed transcribers to write down all disfluencies, such as hesitations, pause fillers, and restarted speech, but if a passage required repeated listening, they were told to make a decision and keep moving. Word fragments were transcribed, though to save time, we dropped the practice of having transcribers guess which word was being spoken in the fragment. For mispronounced words, transcribers wrote their guess at the intended word with a "*" in front of it.

We also made modest changes to the usual rules for transcription of non-speech. We asked transcribers to use "[LAUGH]", "[COUGH]", "[LIPSMACK]", and "[SIGH]" for the corresponding noises, and to use "[MN]" for any other type of mouth noise, and "[NOISE]" for all other noises. Transcribers did not indicate when a speaker laughs while saying a word, only if there is laughter separate from words. We also did not transcribe various steady state background or channel noises. We introduced more flexibility for some issues, again to try to increase speed. For instance, the instructions say to use either the shortened form "kinda" or the full words "kind of" depending on the speech, but that transcribers should not agonize over any particular instance because either form is acceptable.

The transcription service we primarily worked with in this effort, WordWave, had in place a number of careful conventions for transcribing punctuation and capitalization. We considered eliminating these instructions to save time, since this information is not used in our acoustic model training, but we decided to keep them since a) they don't slow the transcription process significantly and b) they may be useful in the future for other applications of this data.

3. Automatic Utterance Segmentation

Acoustic training in ASR systems typically requires segmented transcripts in which the words in a particular utterance are associated with a specific audio segment designated by a begin and end time. As discussed in the introduction, considerations of cost and volume led us to use commercially produced transcripts that had no utterance time information. In this section, we describe a process using BBN's Byblos speech recognition system to automatically segment and recover utterance times given the transcript of the complete conversation.

Although the automatic segmentation process has several components, the heart of the process is a method to break the audio into distinct acoustic regions and align individual utterances from the complete transcript to each region. We investigated three variants of this core algorithm and compared the results of each when used in training. In this section we first describe the general segmentation process, and then the specific variants of the process.

3.1. General segmentation process

The first step in segmentation is to generate cepstral feature vectors that can be used with Byblos' recognition models [7]. The cepstral analysis is normalized for each conversation side to remove the mean and covariance of that side. We wanted to avoid computing the mean and variance statistics over the long silences that can occur in unsegmented conversational speech, since these statistics would not match those seen in training Byblos models, where segmented data was used. We therefore ran a simple speech detector and computed the mean and covariance only on speech regions found by the detector. (This speech detect segmentation was not used in later phases of processing, since it had no information about which words were spoken in any region.)

The next step in the segmentation process is to generate conversation-specific and speaker-specific language models by training n-gram models on the transcriptions for the conversation and for the speaker-side respectively (we assume each conversation side corresponds to a single speaker). We smooth these n-gram models by adding large amounts of training data from a background corpus using a small weight[8]. We also create a conversation-specific decoding dictionary that uses words that appear in the transcripts as well as 1000 frequent words from the background corpus.

The next step is to find the correct channel to speaker correspondence. (This is needed because transcribers are not completely consistent in how they label the two audio channels as left and right). To do this, we decode each channel (side) of a conversation using the Byblos recognizer with the conversation-specific language model. We then compute a 2x2 word error rate matrix by aligning the recognition hypothesis for each channel to the fast transcriptions for each of the sides (speakers). We assign a speaker to a channel by picking the channel whose decoding output has the lowest alignment error for the speaker. Any conversation in which both speakers are assigned to the same channel is discarded.

The next step is to actually segment the audio and assign utterance transcripts to each segment. We investigated three different methods to accomplish this, and these are described in section 3.2, below.

The final step following this is to filter out bad segments. Here, we regenerate the feature vectors for a side using the segments found in the preceding step, and decode the segmented utterances using a speaker-specific language model. We then align the recognition output for each utterance with its putative transcript and discard all utterances with a high error rate. This filtering step rejects utterances with segmentation errors and/or transcription errors. The non-rejected utterances with their associated transcripts are the final output of this process.

3.2. Three segmentation variants

Within the general segmentation process described above, we investigated three specific methods to segment and align the data. In the following, we refer to them as Algorithms I, II, and III.

In Algorithm I, we perform recognition using a biased language model, and we define utterance endpoints at places in the signal where the recognizer finds long silences. This approach provides robustness to noise in the speech
signal, in that such problems will only briefly confuse the recognizer, allowing it to get back on track for the rest of the conversation. The potential weakness of this approach is that the recognizer can make enough errors that utterances in the transcripts end up incorrectly aligned to the speech or discarded.

In Algorithm II we investigated using the recognizer to perform just a coarse initial segmentation of the transcript for a side (i.e. segmenting into longer regions than in the first method), and then performing forced alignment of the words within each segment. In this case the final segments are created by breaking the speech at long silences found either by the recognition pass or in the forced alignment. This method tries to reduce recognition errors by relying on the recognizer for fewer decisions, i.e., only for the coarse segmentation.

The last segmentation approach, Algorithm III, was simply to run forced alignment of a complete transcript side against the corresponding complete audio side, and again to segment utterances at long silences. This method avoids recognizer errors completely, but is susceptible to forced alignment failure if the transcript is incorrect or if there is significant noise in the audio. In fact, our initial attempts to use such a forced alignment failed for a large number of conversations for just such reasons. We therefore modified the basic forced alignment to a) reduce the search pruning to avoid some alignment failures and b) run preprocessing to detect and remove very long silences before running forced alignment. The latter change was introduced after we discovered that our system had significantly worse alignment performance in the middle of long silences that occur when one speaker is listening to the other for an extended period.

4. Feasibility Experiments

We ran experiments to investigate the quality of both the quick transcriptions and the various automatic segmentation methods described in the previous section. In the experiments, we trained recognition models for the Byblos system using a common set of roughly 20 hours of data from the Switchboard-I corpus but varying the transcripts and segmentation used with it. The acoustic models trained were maximum likelihood, gender independent, with VTLN and heteroscedastic linear discriminant analysis (HLDA) applied in feature extraction. The models were tested by running unadapted recognition on the Eval01 test corpus and comparing word error rates (WER). The language model was fixed across all experiments and was trained on the full set of Switchboard transcriptions.

The baseline experiment was to train acoustic models on the 20 hours of data using the very careful manual segmentation and transcription produced by MSU [6]. The LDC made available another quick transcription of the same data set, produced using tools that supported automatic segmentation followed by rapid transcription of the segments. (Note that these transcripts and segments are different from the careful transcripts that LDC originally produced for the Switchboard corpus). We engaged the commercial transcription agency WordWave to provide quick transcriptions of the same 20-hour set with transcription conventions as specified in Section 2, and we segmented these transcripts using the three algorithms described previously. These transcripts were produced at a rate of approximately six times real-time, nearly an order of magnitude faster than traditional, more careful CTS transcription efforts.

Table 1 shows the amount of training and the recognition WER for the above experiments. In the baseline MSU result in the first row, the manual segmentation produced 23.4 hours of actual data and the Eval01 WER was 38.0%. The second row shows results using LDC's quick transcriptions; here the amount of useable training data is reduced to 17.9 hours and the WER increases to 39.4%. We see in the following two rows that Algorithms I and II have similar behavior, with both showing reasonable performance, though still significantly worse than the careful MSU baseline. Finally, the forced alignment-based Algorithm III is shown in the last row. It retains more of the training data than the other quick transcriptions and also has a WER that is only insignificantly worse than the baseline result based on careful MSU transcripts. This final result leads us to conclude that inexpensive transcription is adequate for this task and that forced alignment is a reasonable approach for time segmentation.

Table 1: Word error rates (WER) on Eval01 test set with systems trained on Switchboard-I data.

<table>
<thead>
<tr>
<th>Transcripts/Segmentation</th>
<th>Training hrs</th>
<th>% WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSU</td>
<td>23.4</td>
<td>38.0</td>
</tr>
<tr>
<td>LDC quick transcripts</td>
<td>17.9</td>
<td>39.4</td>
</tr>
<tr>
<td>WordWave/Alg. I</td>
<td>19.5</td>
<td>38.8</td>
</tr>
<tr>
<td>WordWave/Alg. II</td>
<td>19.5</td>
<td>38.8</td>
</tr>
<tr>
<td>WordWave/Alg. III</td>
<td>21.2</td>
<td>38.1</td>
</tr>
</tbody>
</table>
the transcripts and BBN processing the transcripts and segmenting the conversations into utterances before distributing the data to the community through the LDC.

<table>
<thead>
<tr>
<th>AM Training</th>
<th>LM Training</th>
<th>Swbd %WER</th>
<th>Fisher %WER</th>
<th>Overall %WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Baseline</td>
<td>34.5</td>
<td>24.9</td>
<td>29.9</td>
</tr>
<tr>
<td>Baseline</td>
<td>+Fisher</td>
<td>34.0</td>
<td>24.5</td>
<td>29.4</td>
</tr>
<tr>
<td>+Fisher</td>
<td>+Fisher</td>
<td>33.0</td>
<td>23.6</td>
<td>28.5</td>
</tr>
</tbody>
</table>

Table 2: Unadapted word error rates (WER) on the EARS RT03 evaluation set for systems trained with and without 150 hours of quickly transcribed Fisher acoustic and language model training data.

We tested the quality of the first 150 hours of quickly transcribed and segmented Fisher data by adding it to the 370 hours of Switchboard training used to create BBN’s English 2003 conversational telephony speech evaluation system [7]. The models were tested in unadapted decoding on the 2003 EARS evaluation test set, which consists of 3 hours of test data from the Switchboard-II corpus and 3 hours from the Fisher corpus. Table 2 shows that the new data contributes a significant improvement to the overall WER through better acoustic and language models. In addition, both the matched-domain Fisher subset of the test data and the mismatched Switchboard subset appear to improve equally, indicating that the gain comes not just from adding in-domain training.

In addition to the above results, other research sites in the EARS project have begun to use this data. Although this work is still preliminary, several sites have reported significantly reduced WER in their state-of-the-art systems by adding Fisher training data transcribed and segmented using the methods described in this paper.

6. Conclusions

In this paper, we have demonstrated that quickly transcribed data, produced at a fraction of the traditional cost for CTS transcripts, can be used to significantly improve acoustic and language models for conversational speech. On a 20-hour training set, the combination of quick transcriptions with an automatic segmentation algorithm produced time-marked transcripts that perform as well in ASR training as very careful manual transcripts and segmentation. We demonstrated the feasibility of rapidly transcribing a large training corpus quickly and at relatively low cost and found significant gains in WER using this data in preliminary experiments with a state-of-the-art ASR system. In future work we plan to exploit this data to improve CTS recognition further.

7. References