Segmentation of Recordings Based on Partial Transcriptions

Patrick Cardinal, Gilles Boulianne, Michel Comeau

Centre de Recherche Informatique de Montréal
100-550 Sherbrooke street West
Quebec, Montreal, Canada

{patrick.cardinal,gilles.boulianne,michel.comeau}@crim.ca

Abstract

In this paper, we present the approach we used to produce a training database from a set of recorded newscasts for which we had inaccurate transcriptions. These transcribed segments correspond to a set of prepared anchor texts and journalist stories, not necessarily in chronological order of their actual presentation. No segmental time boundary information is provided. Our main concern is thus to establish time marks that delimit the audio segments of the corresponding texts. To resolve this problem, we have developed a time marking procedure using our speech recognition engine. We obtain a segmentation accuracy of 80%.

1. Introduction

The adaptation of a speech recognition system to a new domain requires large amounts of transcribed utterances which are very costly to produce manually. To tackle this problem, Lamel [1] and Nguyen [2] use available closed-captions of television programs as acoustic training data for their speech recognition system. The audio segment boundaries of the corresponding training transcriptions are provided in the closed-captions.

Our situation is quite different. We have a set of prepared anchor texts and journalist stories associated to newscast recordings. However, we have no information whatsoever concerning the segmental time boundaries required for segmenting the recordings into acoustic training data. We have thus developed an alignment procedure that determines the time marks of the audio segment boundaries with the corresponding texts. In addition, we have to take into account the following considerations:

- The audio file and text lengths do not correspond
  The audio may contain untranscribed content. For example, recordings usually contain weather reports and commercials that have no associated transcriptions. Thus we must deal with texts that are sparsely distributed over the recording.

- Transcribed texts are inaccurate
  These texts are often only a guideline for the news reader or journalist. Thus, the text does not correspond exactly to what is said.

- A text may not be in the recording
  For various reasons, an intervention can be cancelled during the newscast and will consequently be absent in the recording. Also, some texts are simply directives to be observed by the news reader.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the approach we used to segment the recording into transcribed audio files. In Section 4, the experimental setup is outlined. Results are presented in Section 5 and finally, the conclusion is given in Section 6.

2. Related Work

A related work is presented in [3] by Columbia University. The investigated problem is to correct the approximate time marks occurring in the closed-caption transcriptions. They arrived at a satisfactory solution by aligning the closed-captions with the audio transcriptions generated by a speech recognizer. Their procedure then reassigns the timecode of each closed-captioned word according to its occurrence in the automatic transcription. The success of their approach depends on closed-captioned time marks that lag in a reasonable range, a couple of seconds in general. This avenue cannot be pursued in our case since no timecode information is available to restrict the search domain.

In [4], CMU describes a story segmentation and commercial detection procedure. They use features extracted from the video, the audio signal and the closed-captions for hypothesized story boundaries in a newscast recording. One part of their work consists in aligning closed-captions with an automatically generated transcription to establish transcription (provided by closed-captions) timecodes. However, specific results concerning alignment accuracy are not mentioned and contrary to our work, they cannot handle a set of texts for which the chronological order is unknown.

The work presented in [5] is another example making use of an alignment algorithm to estimate timecodes of a text. The author uses time provided by closed-captions to mark nearly perfect transcriptions of television programs. But again, timecodes are assumed to be provided with the captioning.

3. Approach

Our starting point is a long audio file (typically half an hour or more) and a set of one or more texts, each of which may or may not be associated with an audio segment. The segmentation process is actually quite simple and is outlined in Figure 1.

There are four modules. The speech recognition module produces a time marked transcription of the audio file. The alignment module aligns each text with the transcription which in turn establishes time boundaries of texts. The analysis module examines each time-marked text to hypothesize its acceptance or rejection. Finally, we can adapt the acoustic model using the resulting segmentation in order to improve the results in a subsequent pass.
3.1. The speech recognition module

This module is our speech recognition system which is based on weighted finite-state transducers (WFSTs) [6]. These have been extensively described by Mohri [7]. Our speech recognizer is used to generate a time-marked transcription of a newscast recording.

3.2. Alignment

In general, the alignment of two strings is obtained by computing their edit-distance which is the minimal cost required to transform one string into the other according to the operations of insertion, deletion, substitution and match [8]. Although dynamic programming is a well known technique to resolve this kind of problem, our alignment implementation uses FSTs [9]. Accordingly, the two strings $s_1$ and $s_2$ are represented by transducers $T_{s_1}$ and $T_{s_2}$ and the edition function is implemented by $T_{edit}$. Figure 2 shows the transcription produced by the speech recognizer and the text to align, represented as transducers.

The construction of the FST representing the transcription is straightforward. However, note that only time markers associated with silence segments are taken into account. This choice is motivated by the assumption that the segments are separated by a silence in the audio.

The set of all possible alignments is computed using the composition operation as shown in the following equation:

$$T_{align} = T_{s_1} \oplus T_{edit} \oplus T_{s_2}$$

where $\oplus$ denotes the operation of composition. The best alignment is extracted from the resulting FST by performing a best path search such as Dijkstra’s algorithm [10]:

$$alignment = BPS(T_{align})$$

where $BPS$ is a best path search function.

Since no time information is available concerning the position of texts in the transcription, the alignment must be computed using the complete recording. With the standard distance function, the algorithm will tend to scatter the words of the text throughout the transcription. This situation is handled by an insertion (of transcription word into the text) cost defined by the following function:

$$cost = \begin{cases} \phi & \text{while the first text word has not been aligned} \\ \varphi & \text{during the alignment of the remaining words} \\ \phi & \text{after last word of the text has been aligned} \end{cases}$$

where $\phi < \varphi$ are the insertion costs. Figure 3 shows the FST edit-distance implementation including this cost function.

The output of this procedure is a set of time-marked alignments.

3.3. Segmentation and analysis

We stated that the input texts may contain errors such as incomplete transcriptions (e.g., in the case of an interview where only the questions are in the text); also a given text may have no associated audio segment. This situation may arise when an
intervention is cancelled at broadcast time. This module uses alignment statistics for filtering out segments. A segment is rejected if:

The number of words in the text is less than a threshold \( t_w \)

Too few words are indicative of noise, a directive to be observed by the news reader or an incomplete text.

The number of matched words is less than a threshold \( t_m \)

A low number of matched words may indicate that the text has not been spoken or is incomplete.

The number of deletions is greater than a threshold \( t_d \)

This rule is used in conjunction with the previous one in order to intercept incomplete or non-spoken texts.

Chronological order analysis (COA) This rule is active only when it is known in advance that the texts are in the chronological order of their actual presentation. This filter allows us to reject a segment whose boundary times are in contradiction with its assumed neighbors.

4. Experimental Setup

This section describes the context in which the experiments have been made.

4.1. Acoustic Models

The primary motivation of this work is the implementation of a lightly supervised approach for adapting our speech recognition system to a new domain. To simulate such a situation, we use a reading speech model trained in a clean environment applied to our present problem, that involves spontaneous speech in a noisy environment.

Sentences used for acoustic training come from La Presse and Hansard (Canadian parliamentary corpus). Training sentences were recorded in a reading speech style by 95 Canadian French speakers (49 men, 46 women), in a quiet environment. A sampling rate of 16 kHz was used. The resulting audio database contains about 36 hours of continuous read speech.

Acoustic data were analyzed into 39 parameters (12 MFCCs including log-energy along with the first and second order derivatives). Speaker-independent acoustic models were initially trained using HTK. Resulting cross-word triphone models are represented by 1588 tied-state distributions with 16 Gaussians per state.

The model has also been augmented with 6 English phonemes to take into account English pronunciations of some words.

The baseline recognition vocabulary contains 20K words for which base pronunciations including liaison were automatically obtained using a set of grapheme-to-phoneme rules.

4.2. Language Model

For the training of the generic language model, we used generic data from newspapers published in Québec (La Presse and Le Soleil), and in-domain data from TVA’s older news archives (NewsView) and TVA’s more recent line-up archives (TVA archives). Altogether these corpora include 144 M words and the most recent 100 K words were held-out from training to provide a development set.

We selected a 20K word vocabulary from the most frequent words of a weighted combination of the corpora, with weights chosen to minimize the out-of-vocabulary rate on the TVA development set. The vocabulary has been enriched with 10K new unigram probabilities and the language model was adapted using [11].

The generated language model contains 29237 unigrams, 296479 bigrams and 202994 trigrams and has a perplexity of 79 on the test set.

4.3. Corpus Description

The corpus used for our experiments consists in 32 (809 texts) newscast recordings of average duration one half hour, broadcast by the Québec television network TVA between July 8th and August 19th, 2002. 27% of these recordings are commercials for which no text is available. Recordings also contain musical segments, often superimposed with speech.

Five of these recordings (122 texts) constitute our development set; the 27 remaining recordings represent our test set.

5. Experimental Results

Two metrics are used to evaluate the performance of our procedure. The first one assesses the quality of the segmentation. A text is deemed to be accurately aligned if its beginning time falls after the previous segment’s real ending, and before the following segment’s real beginning. We define the segmentation accuracy as:

\[
ACC = \frac{\text{Number of correctly aligned segments}}{\text{Number of accepted segments}} \times 100
\]

The second metric combines the false rejection rate (FRR) and false acceptance rate (FAR). A false rejection occurs when a segment is not in the final segmentation set but should be. False acceptance occurs when a segment is in the final segmentation set but should have been rejected. The false rejection and acceptance rates are defined as:

\[
FRR = \frac{\text{Number of false rejections}}{\text{Total number of segments}} \times 100
\]

\[
FAR = \frac{\text{Number of false acceptances}}{\text{Total number of segments}} \times 100
\]

The first experiment examines the effect of the \( t_d \) and \( t_m \) values on the accuracy, false rejection and acceptance rates. For this experiment, the chronological order analysis was deactivated.

![Accuracy vs False Rejection Rate](image)

Figure 4: Accuracy vs False Rejection Rate.

Our results show that the false acceptance rate is not influenced by the threshold values. Figure 4 shows that a filter based
on the number of matched and deleted words influences the performance of the segmentation procedure. Thus, the threshold values may be optimized for the specific needs of an application.

For the following experiments, the threshold values have been chosen to maintain the false rejection rate at a reasonably low value in the development set \( t_e = 10, t_m = \text{#words}/2, t_d = \text{#words}/6 \).

The second experiment examines the effects of the different elements of the analyser, the acceptance and rejection rates. Applying the procedure without the analyser is the baseline of this experiment. We have first tested the effect of the chronological analysis only. Then, we have tested the effect of the threshold values. Finally, we have combined the COA with thresholds. The results are shown in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>% Acc</th>
<th>FRR</th>
<th>FAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-No filter</td>
<td>59.00%</td>
<td>3.62%</td>
<td>18%</td>
</tr>
<tr>
<td>2-COA</td>
<td>61.7%</td>
<td>8.59%</td>
<td>7.03%</td>
</tr>
<tr>
<td>3-( t_e )</td>
<td>59.00%</td>
<td>3.62%</td>
<td>16.67%</td>
</tr>
<tr>
<td>4-( t_e + t_m )</td>
<td>65.79%</td>
<td>21.17%</td>
<td>8.03%</td>
</tr>
<tr>
<td>5-( t_e + t_m + t_d )</td>
<td>80.70%</td>
<td>35.04%</td>
<td>7.30%</td>
</tr>
<tr>
<td>6-( t_e + t_m + t_d + \text{COA} )</td>
<td>80.70%</td>
<td>37.80%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 1: Effect of the analyser on the development set.

This experiment shows that the accuracy and the false rejection rates are mainly influenced by the \( t_e \) and \( t_m \) thresholds as shown by the results of lines 1–4 compared to those of lines 5–6. On the other hand, the chronological order analyser (COA) has a big influence on the false acceptance rate as shown by lines 2 and 6. This result is not surprising since a segment which has boundaries in contradiction with its assumed neighbors is a strong indication that the text is absent in the recording. Moreover, these results show that the procedure performs well even when the chronological order is unknown.

The last experiment examines the effects of the biased language model and the segmentation-based adapted acoustic model of a previous iteration. The baseline of this experiment is a transcription produced with the generic models and all elements of the analyser activated. The results are shown in Table 2. Note that the results shown in table 1 were obtained with MLLR and BLM models.

<table>
<thead>
<tr>
<th></th>
<th>% Acc</th>
<th>FRR</th>
<th>FAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>62.5%</td>
<td>57.48%</td>
<td>0%</td>
</tr>
<tr>
<td>baseline + BLM</td>
<td>64.91%</td>
<td>37.80%</td>
<td>0%</td>
</tr>
<tr>
<td>baseline + MLLR</td>
<td>71.43%</td>
<td>49.7%</td>
<td>0%</td>
</tr>
<tr>
<td>baseline + MLLR + BLM</td>
<td>80.7%</td>
<td>37.80%</td>
<td>0%</td>
</tr>
<tr>
<td>Test set</td>
<td>80.22%</td>
<td>29.24%</td>
<td>0.14%</td>
</tr>
</tbody>
</table>

Table 2: Effect of the acoustic adaptation and the biased language model on the development set.

The results show the beneficial effect of the biased language model (BLM) and the MLLR-adapted acoustic model on the performance of the system. On the development set, we obtained a segmentation accuracy of 80.7%; the test set performs equally well at 80.2%.

It is also worth noting that the segmentation errors (that account for 19.8%) are by no means coarse. Although we have yet to quantify these errors on the overall test set, preliminary inspection of the corresponding segments shows that these errors, that translate into associated text words that are outside the computed boundaries, or boundaries that overlap words of a neighboring segment, involve very few words indeed (typically three or four). Thus, these segments can safely be included in applications such as the lightly supervised learning variants of references [1, 2].

6. Conclusion

We have proposed an approach to produce a training database from a set of recorded audio for which we have inaccurate transcriptions and no available timecode information. The procedure for aligning each text with an automatically generated transcription of the audio has shown to yield promising results.

Two aspects have been studied: the false rejection rate and the segmentation accuracy. Our experiments were applied on a set of newscast recordings with associated prepared anchor texts and journalist stories. We obtain a segmentation accuracy of 80% with a false rejection rate of 30%. Moreover, the underlying filtering process correctly identifies text segments that are not rendered in the recording.

Another application of our procedure could be the creation of synchronized closed-captions from available texts of a television program.

7. References


