Automatic Formatted Transcripts for Videos

Aasish Pappu, Amanda Stent

Yahoo! Labs
{aasishkp, stent}@yahoo-inc.com

Abstract
Multimedia content may be supplemented with time-aligned closed captions for accessibility. Often these captions are created manually by professional editors—an expensive and time-consuming process. In this paper, we present a novel approach to automatic creation of a well-formatted, readable transcript for a video from closed captions or ASR output. Our approach uses acoustic and lexical features extracted from the video and the raw transcription/caption files. We compare our approach with two standard baselines: a) silence segmented transcripts and b) text-only segmented transcripts. We show that our approach outperforms both these baselines based on subjective and objective metrics.

Index Terms: Spoken Language Processing, Closed-Captions, Spoken Text Normalization

1. Introduction
Multimedia content such as video and audio files may be supplemented with closed captions for accessibility. Captioning multimedia content is typically a two step process: 1) transcribe the content to obtain text and non-speech events (e.g., applause), and 2) temporally align the transcription with the content to produce closed captions. Closed captions and transcripts not only make multimedia content accessible, but can also improve the “searchability” of the content [1], assist in video classification [2, 3] and video segmentation [4, 5], and help to highlight salient objects in a video frame [6].

Although closed captions can be very useful, the traditional approach to closed captioning is an expensive and time-consuming process, requiring multiple rounds of manual transcription and alignment. Even after this, while the captions may be accurate, manual time alignments are typically perceptibly “off”. In addition, most search engine operators do not index closed caption files, but only index text made visible in a web page, so in order for multimedia content to be treated as “first-class” web content, it must be accompanied by visible transcripts. Most content publishers are unwilling to accompany their videos with poor-quality transcripts such as might be produced by simply printing out raw transcriptions or closed captions; that is, transcripts must be readable and not look “spammy” (e.g., big blocks of text, long sentences). However, manual construction of a readable and well-formatted transcript requires additional time-consuming and expensive rounds of editing following closed captioning.

In this paper, we present a system that takes a raw transcription of a video (e.g., crowd-sourced or ASR output) as input and generates: (i) accurately time-aligned closed captions; and (ii) a readable, well formatted transcript with punctuation, capitalization, and paragraph segmentation. Because the manual effort is reduced to straight transcription, considerable time and money can be saved and more multimedia can be made accessible and searchable. Our system has four components: alignment of transcript and video, punctuation insertion, capitalization and paragraph segmentation. In this paper we focus in particular on the task of punctuation insertion. We demonstrate that a combination of textual and acoustic features leads to higher accuracy on this task than either feature type alone, and that this leads to better decisions in later stages of the transcript formatting pipeline (capitalization, paragraph break insertion).

This paper is organized as follows. First, we present a brief overview of previous work on formatting raw speech transcripts. Then, in Section 3 we describe our system. In Section 4 we present evaluations of our system, focusing on punctuation insertion and its downstream impacts. Finally, we make concluding remarks and briefly describe future directions for this work.

2. Related Work on Punctuation Insertion
There has been a lot of previous work specifically on punctuation insertion in speech transcripts. Here we mention only highlights. Previous work on formatting speech recognition output has shown that both textual and frame-level features can be useful for punctuation prediction. Huang and Zweig used lexical and pause features in a maximum entropy tagger trained and tested on Switchboard conversational speech [7]. Kim and Woodland used lexical, pause, F0 and RMS features in a decision tree framework [8]. Liu et al. were the first to apply conditional random fields to this task [9]. In all cases they looked only at insertion of commas, question marks and periods. Other researchers have obtained good results for both punctuation and capitalization using only lexical information, with large quantities of training data [10, 11, 12]. More recent work has shown that accurate punctuation prediction can improve machine translation output with cross-lingual features [13, 14, 15]. In this work we show that functional features of low-level acoustic descriptors can complement textual features in punctuation of raw transcripts, and that improvements here can lead to outsize impacts on downstream processing (e.g., capitalization), and improved overall readability of final formatted transcripts.

3. System
Our system takes as input a video and a raw transcription of the speech in the video. This transcription may be obtained through automatic speech recognition or (with considerably higher accuracy) through crowdsourcing or professional transcription [16]. The input is processed in four stages.

3.1. Preprocessing and Alignment
We extract the audio from the input video. We obtain a phoneme-level transcription from the input word-level tran-
scription using the CMU pronunciation dictionary\(^1\) and SE-QUITUR \(^1\) for out of dictionary words. Then, we run the P2FA forced aligner \(^3\) with the HUB4 acoustic model from the Sphinx open-source speech recognizer \(^1\) on the audio and phoneme-level transcription. The process is outlined in Figure 1. The result is timestamps for every word in the input transcription, as well as for silences (regions of no speech). Figure 2 shows an example time-aligned speech segment followed by silence. We use this time-aligned data in our speech-based punctuation insertion model. As a side effect, we can also construct closed captions from the time-aligned transcription.

3.2. Punctuation Insertion

We use the frontend from the Flite text-to-speech synthesizer \(^2\) to normalize and homogenize the raw transcription. We run the normalized transcription through a punctuation insertion system trained on a corpus of well-formatted video transcriptions. We implemented systems for performing punctuation insertion using textual and acoustic information alone, as well as an ensemble approach (see Section 4.2.4).

3.3. Capitalization

After inserting punctuations into the transcription, we extract sentences. Within each sentence, we capitalize tokens wherever applicable i.e., named entities, tokens at the beginning of a sentence, and other special cases such as I’m. We use the recaser tool in the MOSES machine translation toolkit \(^2\), trained on one year of news articles, for capitalization.

3.4. Paragraph Boundary Insertion

There has been relatively little work on paragraph break insertion, and most existing methods require considerable processing, e.g. parsing \(^3\). In a large-scale commercial video transcription system there is very little time for preprocessing. To group sentences into paragraphs, we use the TextTiling algorithm \(^3\). This algorithm allows us to detect topic shifts across sentences and insert paragraph breaks when topic shifts occur. A topic shift is detected by computing lexical similarity between adjacent groups of sentences. When new words are introduced in a group of sentences, the algorithm attempts to insert a paragraph boundary preceding this group. The number of paragraph boundaries is determined by the distribution of the topic shift scores across the whole text.

4. Experiments

4.1. Data

We trained and evaluated our system using a set of videos that we obtained from Yahoo Screen. The videos cover several genres — finance, sports, odd news, SNL, comedy, and trending now. The video lengths range from 20 seconds to 12 minutes. The primary language used in all videos is English. For each video, we have high quality, professionally created closed captions containing punctuation and capitalization (but no paragraph boundaries). We split the data into training data (243 videos), development data (121 videos) and test data (35 videos), randomly but balancing across genres. We trained models for both text-based and speech-based punctuation insertion on the training data, and tuned the parameters for the ensemble model using the development data.

4.2. Punctuation Insertion

First, we compare models based on textual and speech features against standard baselines. Then, we compare these models with an ensemble method.

4.2.1. Baselines

We compare our models against the following baselines:

- **Baseline1**: Uses a trigram language model to insert punctuation. We trained the language model on a corpus of news articles. This baseline can insert all types of punctuation, and is similar to the phrase-break insertion approach used in speech synthesis \(^3\).

- **Baseline2**: Uses the silence durations between phrases to insert punctuation according to: \((\text{min} \_ \text{silence} < \text{comma} < \text{max} \_ \text{silence} < \text{period})\). We used \(\text{min} \_ \text{silence} = 0.22\) and \(\text{max} \_ \text{silence} = 0.4\), which we computed based on 10-fold cross validation over the training data. This baseline can only insert \{comma, period, none\}.

4.2.2. Text-Based Model

We use the CRF++ toolkit \(^4\) to train a sequence tagger to insert all types of punctuation (including none) between each pair of words in the input transcription. As features, we use part-of-speech (POS) tags and tokens from the transcription. We use the CLEARNLP toolkit \(^3\) with its off-the-shelf model to predict POS tags.
4.2.3. Speech-Based Model

Before we describe the speech-based punctuation model, we would like to provide the intuition behind using acoustic features to insert punctuation. Punctuations in spoken language not only serve as phrase breaks but also convey the sentiment/emotion of the speaker. For example, in Figure 2, one may predict from the transcription that the text should end with a PERIOD, but based on the pitch contour, which shows a sharp rise and fall in intonation, one may predict an EXCLAMATION. Therefore, we treat this problem as a hybrid of phrase-break prediction and emotion detection.

Emotion detection in speech is a well-studied problem (e.g. [27,28]). It is known that functional features are critical in these tasks [29]. A functional feature takes a sequence of low-level feature descriptors as input and produces a fixed-size vector as output. Since functional features are computed over low-level descriptor contours, they capture trends over the entire speech segment. Since the length of the output vector is independent of the length of the input sequence, we can easily perform feature selection across the vector’s dimensions.

Our speech-based punctuation insertion system operates only on silences from the input time-aligned transcription. We used openSMILE [30] to extract 12 functional features (listed in Table 1) over the speech segment preceding each silence. We compute these functional features over four low level feature descriptors: Energy, Pitch Onsets, and Duration. The feature extraction process results in a 2268 dimensional vector, which we use to classify each silence as one of {exclamation, questionmark, period, comma, hyphen, none}. We use the Weka [31] implementation of Random Forests with 30 trees and maximum depth computed based on cross-validation on the training data.

4.2.4. Ensemble Method

We hypothesize that textual or speech features alone may provide incomplete information; for example, silences may not always translate into punctuations in the text and vice-versa. Table 2 shows silences in the training data that map to punctuations and silences that do not. We notice that only 1/3 of silences correspond to punctuations. To incorporate textual and speech information, we trained an ensemble method, a logistic regression model trained over the development data using the predicted labels and confidence scores from the text based and speech based models.

### Table 1: Functional features used in speech based punctuation insertion

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td></td>
</tr>
<tr>
<td>Pitch Onsets</td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: Silences vs. punctuations

<table>
<thead>
<tr>
<th>Method</th>
<th>WER</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline1</td>
<td>18.7</td>
<td>72.2</td>
</tr>
<tr>
<td>Baseline2</td>
<td>14.4</td>
<td>87.8</td>
</tr>
<tr>
<td>Text only</td>
<td>9.7</td>
<td>92.4</td>
</tr>
<tr>
<td>Speech only</td>
<td>10.3</td>
<td>92.4</td>
</tr>
<tr>
<td>Ensemble</td>
<td>9.1</td>
<td>93.6</td>
</tr>
</tbody>
</table>

### Table 3: Results: punctuation insertion

4.2.5. Results

We evaluate these methods against the professionally created closed captions for each video, which include punctuation. We use F1 score to assess label accuracy and word error rate (WER) to assess label positioning. Table 3 shows results for our two baseline methods, our text and speech based models, and the ensemble method. We observe that both individual models outperform the baselines. In addition, the ensemble model outperforms all the other methods on both metrics. We conclude that textual and speech features complement each other for this task.

4.3. Capitalization

Sentence-final punctuation is crucial for capitalization. The closed captions in our data include capitalization, allowing us to measure capitalization accuracy for different punctuation methods. As Table 4 shows, the ensemble method outperforms other methods by 7%. This outsize performance difference is due to better punctuation with the ensemble model, particularly with respect to end-of-the-sentence punctuations (period, questionmark and exclamation).

### Table 4: Results: capitalization

<table>
<thead>
<tr>
<th>Method</th>
<th>WER</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text only</td>
<td>65.2</td>
<td></td>
</tr>
<tr>
<td>Speech only</td>
<td>64.0</td>
<td></td>
</tr>
<tr>
<td>Ensemble</td>
<td>71.1</td>
<td></td>
</tr>
</tbody>
</table>
Table 6: Difference between automatic and manual alignments. ← means automatic segments are misaligned. → means manual segments are misaligned.

<table>
<thead>
<tr>
<th>Segment Type</th>
<th>Instances</th>
<th>Begin (sec)</th>
<th>End (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>1812</td>
<td>→6.24</td>
<td>→6.1</td>
</tr>
<tr>
<td>Non-speech</td>
<td>44</td>
<td>←22.62</td>
<td>←21.57</td>
</tr>
</tbody>
</table>

5. Conclusions and Future Work

We describe a system for generating well-formatted transcripts for videos from input raw transcriptions/closed captions. The system includes components for alignment of transcription to video, punctuation insertion, capitalization, and paragraph break insertion. We focus on the task of punctuation insertion, as it is critical to later stages. In both quantitative and qualitative evaluations, we show that an ensemble method combining acoustic and textual features outperforms speech-based and text-based methods, and leads to outsize improvements in later stages of processing.

Although we achieve high accuracy for punctuation insertion, we still miss 7% of punctuations. We see that acoustic features are inaccurate when alignment fails due to non-speech segments. In addition, our system currently does not detect speaker change events, which should generally correspond to punctuation insertions. In future work, we could add speaker change detection, acoustic event detection and music identification to the preprocessing and alignment stage of our system.

6. References


