



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 accessi-

ble 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-

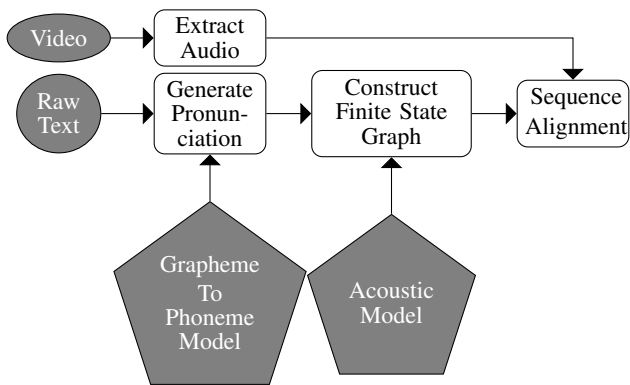


Figure 1: Workflow to obtain time-aligned transcriptions

scription using the CMU pronunciation dictionary¹, and SEQUITUR [17] for out of dictionary words. Then, we run the P2FA forced aligner [18] with the HUB4 acoustic model from the Sphinx open-source speech recognizer [19] 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 [20] 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 [21], 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 [22, 23]. 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 [24]. 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.

¹www.speech.cs.cmu.edu/cgi-bin/cmudict

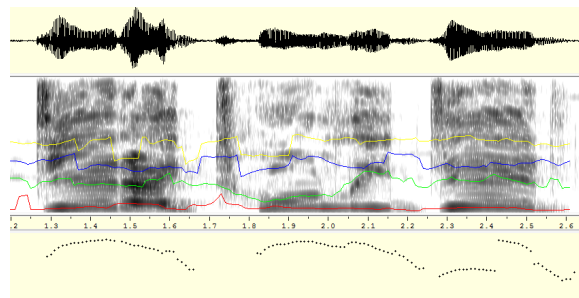


Figure 2: Spectrogram and pitch contour for an example utterance with sharp rise and fall in intonation. Utterance text: *wished for a sandwich i did i totally did <sil>*

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 [25].
- *Baseline2*: Uses the silence durations between phrases to insert punctuation according to: $(min_silence < comma < max_silence < period)$. We used $min_silence = 0.22$ and $max_silence = 0.4$, 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 (`crfpp.sourceforge.net`) 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 [26] with its off-the-shelf model to predict POS tags.

Extremes	max, min, range
Means	arithmetic, geometric
Peaks	num. peaks, distance between peaks
Segments	num. segments
Onset	num. onsets, offsets
Moments	st. deviation, variance
Crossings	zero-crossing rate, mean crossing rate
Percentiles	percentile values, inter-percentile ranges
Regression	linear and quadratic regression coefficients
Samples	sampled values at equidistant frames
Times	rise and fall of the curve, duration
DCT	DCT coefficients

Table 1: Functional features used in speech based punctuation insertion

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, Voicing probability, 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.

	Punctuation	No punctuation
Silence	6659	8169
No silence	6577	-n/a-

Table 2: Silences vs. punctuations

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).

4.4. Subjective Evaluation

Sentence-final punctuation is also crucial for paragraph breaking; however, closed captions do not contain paragraph breaks. We conducted a crowdsourced evaluation using Amazon mechanical turk to assess the overall readability of the transcripts our system can produce, including punctuation, capitalization and paragraph breaks. Turkers were presented with a video (including closed captions automatically aligned by our system) and three transcript variants produced using: (a) text based punctuation insertion followed by capitalization and paragraph break insertion; (b) speech based punctuation insertion followed by capitalization and paragraph break insertion; and (c) the ensemble method for punctuation insertion followed by capitalization and paragraph break insertion. In all cases, the tran-

Method	WER	F1
Baseline1	18.7	72.2
Baseline2	14.4	87.8
Text only	9.7	92.4
Speech only	10.3	92.4
Ensemble	9.1	93.6

Table 3: Results: punctuation insertion

Punctuation model	Accuracy
Text only	65.2
Speech only	64.0
Ensemble	71.1

Table 4: Results: capitalization

Method	Paragraphs	Punctuations	Capitalized words	Macroaveraged rank	Winners	Losers
text only	2.89	16.92	22.69	1.92	6	4
speech only	1.70	7.65	20.36	2.17	6	15
ensemble	3.88	20.12	25.96	1.87	10	1

Table 5: Statistics on the test data and subjective evaluation results

scripts were automatically preprocessed to remove white space between a punctuation and the previous token, remove white space within contractions like *we'll*, and normalize numbers. Turkers were told that the transcript variants were produced by computer programs. They were asked to rank the transcript variants from 1 (best / most readable) to 3 (worst / least readable). They were allowed to assign more than one transcript to a ranking, but were asked to use the whole range of rankings from 1 to 3 if possible. They were also asked to explain what made the best transcript(s) better, and what made the worst transcript(s) worse. We produced one HIT for each of the 35 videos in our test data. Each HIT was assigned to seven turkers. Turkers were required to be US-based (thus, presumably fluent English speakers) and to have completed at least 100 HITs with an approval rating of 90% or higher. We paid \$0.15 per assignment.

Table 5 shows some statistics about the punctuation, capitalization and paragraph breaks in the evaluated transcripts, as well as the macroaveraged results of the subjective evaluation and the number of test documents for which each method was a “winner” (four or more judgments of rank 1) or a “loser” (four or more judgments of rank 3). The ensemble method has highest overall readability. In their comments, turkers praised highly-ranked transcript variants for the flow of the paragraph and sentence breaks, and condemned low-ranked transcript variants for being “chopped up” (too many paragraph/sentence breaks), “one big block of text” (too few paragraph/sentence breaks), missing speaker changes, or missing capitalizations.

4.5. Analysis

We observe that on certain videos even the ensemble system cannot achieve good performance. This is typically due to non-speech events in the videos. In our data, videos contain the following non-speech events: laughter, music, applause and generic background noise. To fix this problem, we could use chroma features to detect music events and use acoustic event datasets² to filter laughter, applause and other noise events.

Since we use acoustic features for punctuation insertion, accurate alignment of text and speech is critical. We compared our automatically-obtained word-level time alignment with the manually-created time alignments available from the professionally created closed captions. Since word-level manual alignments are not available with the closed captions, we only analyzed speech segments and non-speech events. From Table 6, we see that automatically aligned non-speech events are misaligned by 20 seconds, whereas manually aligned speech segments are misaligned by 6 seconds. Automatic alignment does well for speech; however, non-speech segments introduce errors during extraction of speech features due to misalignment.

²c4dm.eecs.qmul.ac.uk/sceneseventschallenge/description.html

SegmentType	Instances	Begin(sec)	End(sec)
Speech	1812	→6.24	→6.1
Non-speech	44	←22.62	←21.57

Table 6: Difference between automatic and manual alignments. ← means automatic segments are misaligned. → means manual segments are misaligned.

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

- [1] J.-C. Shim, C. Dorai, and R. Bolle, “Automatic text extraction from video for content-based annotation and retrieval,” in *Proceedings of the International Conference on Pattern Recognition*, 1998.
- [2] D. Brezeale and D. Cook, “Using closed captions and visual features to classify movies by genre,” *Proceedings of the ACM KDD Workshop on Multimedia Data Mining*, 2006.
- [3] K. Filippova and K. Hall, “Improved video categorization from text metadata and user comments,” in *Proceedings of SIGIR*, 2011.
- [4] A. Hauptmann and M. Witbrock, “Story segmentation and detection of commercials in broadcast news video,” in *Proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries*, 1998.
- [5] T. Cour *et al.*, “Movie/script: Alignment and parsing of video and text transcription,” in *Proceedings of ECCV*, 2008.
- [6] M. Bertini, C. Colombo, and A. Del Bimbo, “Automatic caption localization in videos using salient points,” in *Proceedings of ICME*, 2001.
- [7] J. Huang and G. Zweig, “Maximum entropy model for punctuation annotation from speech,” in *Proceedings of INTERSPEECH*, 2002.
- [8] J. Kim and P. Woodland, “The use of prosody in a combined system for punctuation generation and speech recognition,” in *Proceedings of INTERSPEECH*, 2001.

- [9] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using conditional random fields for sentence boundary detection in speech," in *Proceedings of the ACL*, 2005.
- [10] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *Proceedings of ICASSP*, 2009.
- [11] W. Lu and H. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of EMNLP*, 2010.
- [12] M. Shugrina, "Formatting time-aligned ASR transcripts for readability," in *Proceedings of HLT/NAACL*, 2010.
- [13] S. Peitz, M. Freitag, A. Mauser, and H. Ney, "Modeling punctuation prediction as machine translation," in *Proceedings of the International Workshop on Spoken Language Translation*, 2011.
- [14] F. Batista, H. Moniz, I. Trancoso, and N. Mamede, "Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 474–485, 2012.
- [15] J. Miranda, J. Neto, and A. Black, "Improved punctuation recovery through combination of multiple speech streams," in *Proceedings of ASRU*, 2013.
- [16] R. Kushalnagar, W. Lasecki, and J. Bigham, "A readability evaluation of real-time crowd captions in the classroom," in *Proceedings of ASSETS*, 2012.
- [17] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.
- [18] J. Yuan and M. Liberman, "Speaker identification on the SCOTUS corpus," in *Proceedings of Acoustics '08*, 2008.
- [19] P. Placeway *et al.*, "The 1996 Hub-4 Sphinx-3 system," *Proceedings of the DARPA Speech Recognition Workshop*, 1997.
- [20] A. Black and K. Lenzo, "Flite: a small fast run-time synthesis engine," in *Proceedings of the ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.
- [21] P. Koehn *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the ACL*, 2007.
- [22] K. Filippova and M. Strube, "Using linguistically motivated features for paragraph boundary identification," in *Proceedings of EMNLP*, 2006.
- [23] C. Sporleder and M. Lapata, "Broad coverage paragraph segmentation across languages and domains," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 3, no. 2, 2006.
- [24] M. Hearst, "TextTiling: Segmenting text into multi-paragraph subtopic passages," *Computational linguistics*, vol. 23, no. 1, pp. 33–64, 1997.
- [25] P. Taylor and A. Black, "Assigning phrase breaks from part-of-speech sequences," *Computer Speech and Language*, vol. 12, pp. 99–117, 1998.
- [26] J. Choi, "Optimization of natural language processing components for robustness and scalability," Ph.D. dissertation, 2012.
- [27] B. Schuller, S. Steidl, and A. Batliner, "The INTERSPEECH 2009 emotion challenge," in *Proceedings of INTERSPEECH*, 2009.
- [28] M. Valstar *et al.*, "Avec 2013: the continuous audio/visual emotion and depression recognition challenge," in *Proceedings of the International Audio/Visual Emotion Challenge and Workshop*, 2013.
- [29] B. Schuller *et al.*, "The INTERSPEECH 2010 paralinguistic challenge," in *Proceedings of INTERSPEECH*, 2010.
- [30] F. Eyben *et al.*, "Recent developments in openSMILE, the Munich open-source multimedia feature extractor," in *Proceedings of ACM Multimedia*, 2013.
- [31] M. Hall *et al.*, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.