Automated Closed-Captioning Using Text Alignment

Anthony F. Martone, Cuneyt M. Taskiran, and Edward J. Delp

Video and Image Processing Laboratory (*VIPER*)
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana

ABSTRACT

The production of closed captions is an important but expensive process in video broadcasting. We propose a method to generate highly accurate off-line captions efficiently. Our system uses text alignment to synchronize program transcripts obtained for a video program with text produced by an automatic speech recognition (ASR) system. We will also describe the accuracy in both closed-caption text and the ASR output for a number of news programs and provide a detailed analysis of the errors that occur.

Keywords: closed caption generation, text alignment, off-line captions, automatic speech recognition, accuracy of captions

1. INTRODUCTION

The importance of closed captioning is constantly growing due to strict Federal Communication Commission (FCC) regulations. In 1993 the FCC mandated that all television sets with screens of 13 diagonal inches or larger that are sold in the United States must contain closed caption decoders. Currently the FCC requires that 900 programming hours per channel per quarter be closed captioned for every television station. By 2006 all television programs, with few exceptions, must be captioned.

Current closed-captioning technology consists of two approaches: online and off-line. Online closed-captions are generated in real time by an operator watching the program. These captions are error prone, have a delay of a few seconds, and do not have a well-segmented speaker and sentence structure. The quality of off-line captions is much higher. The captions are well-aligned and have no delay. However, they are more costly to generate.

One alternative approach to real-time captioning is known as electronic newsroom captioning (ENC). In this method, prior to the airing of the program, program scripts are entered into the newsroom computer network, which is often referred to as an electronic newsroom. As the teleprompter operators advance the script on the prompter monitor, they also control the transmission of the script to the caption encoding device. Although much cheaper to produce than real-time captioning, ENC poses some problems. Since only material that is scripted can be captioned with this technique, breaking news, sports and weather updates, and live field reports cannot be captioned. Another problem is that the optimum display speed for prompted text may be too fast for closed-caption text displays.

Direct use of automatic speech recognition (ASR) technology could be used to generate both real time and off-line captions. However, the accuracy for current ASR systems may not be able to achieve the acceptable accuracy of closed-caption text. In this paper the accuracy of both text generated by an ASR system and closed captioning text are compared. It will be shown that the accuracy of current ASR technology falls below the accuracy of closed-caption text; therefore an ASR system can not be used as a standalone system to produce closed-caption text. We propose an alternative approach to off-line captioning. The components of the proposed system are illustrated in Figure 1. An ASR system is used to process an input video sequence and create a text transcript with time code. A program transcript, produced by a human operator, is then aligned with the ASR output to obtain a highly accurate transcript of the program with time code. This transcript may then be used as closed-caption text and inserted into the program.

This work was supported by a grant from the C-SPAN Archives. Address all correspondence to E. J. Delp, ace@ecn.purdue.edu, telephone: +1 765 494 1740.

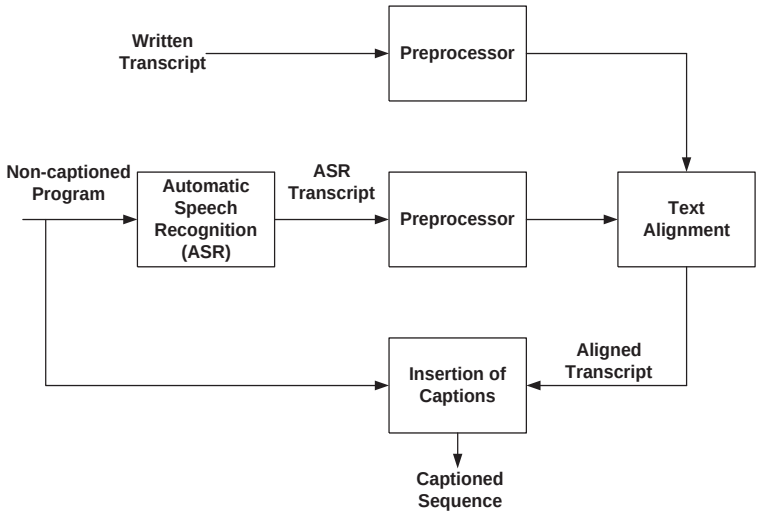

Figure 1. Components of the proposed automated off-line closed-caption text generation system.

2. CLOSED-CAPTIONING TECHNOLOGY

Text added to the video signal of a TV program and displayed in the picture are known as *captions*.[1] *Open captions* are captions that are directly added to the frames of the program and become an integral part of the television picture. A common example of open captions are the names and titles of the speakers appearing in a program, which are displayed in the lower part of the TV screen. The viewer has no control over this type of captions.

Closed captions provide visual text to describe dialogue, background noise, and sound effects in television programs.[2] These captions are hidden in the video signal, invisible without a special decoder. As of July 1993, the FCC mandated that all television sets with screens of 13 diagonal inches or larger sold in the United States must have closed caption decoders. Since we will exclusively deal with closed captions in this paper, we will refer to closed-captions simply as captions throughout the paper.

The two methods used to generate captions are off-line and real-time. In *off-line captioning* a human operator known as the captioner transcribes the audio portion of the program using special captioning software. Depending on the program being captioned, the off-line captioning process can take up to 15 hours per broadcast half-hour. After a final review, the caption text is usually inserted into the video signal prior to broadcast. This is done by synchronizing the video time code with the captions. The time code triggers the corresponding caption data, which is sent to the encoder and inserted into the video on line 21 of the vertical blanking interval. The captioned video is recorded onto a second videotape, which is known as the closed-captioned master. Once the encoding session is finished, the closed-captioned master is ready for re-broadcast, duplication, or distribution with the caption stream (or streams) permanently embedded in the video signal.[1]

In contrast to off-line captioning, *real-time captioning* is done during live programs, such as news, sports, or special live broadcasts. In this method a specially trained stenographer watches or listens to a program while it airs and uses an electronic stenotype keyboard to transcribe the text. Using machine shorthand, a qualified caption writer can keep up with the fastest-speaking news reporters, who often speak at speeds in the range of 225-250 words per minute. The output of the stenotype keyboard is processed using a translation algorithm. The translation algorithm searches large dictionary files, customized to the individual stenographer, which contain stenographic key strokes and the text they represent. When a particular stenotype stroke is found in the dictionary, the algorithm generates the corresponding text and captioning codes. The captions are sent back to the broadcaster, usually via a telephone line, and are encoded into the television broadcast signal. They are displayed line by line on the TV screen, usually with a delay of two to three seconds after the words are spoken. Real-time captions have a larger error rate than off-line captions.

The FCC has prescribed rules and implementation schedules for the captioning of video programming for television broadcasts. All English language programming, which was first shown on or after January 1, 1998, must be captioned over an eight-year period; by 2006 all non-exempt new programs must be captioned.

Analog closed-captioning information is stored in line 21 of the vertical blanking interval, while digital closed captions are inserted in the frame user data fields of the MPEG-2 sequence. For further details about placament and insertion of captions in digital and analog video refer to.[3]

3. TEXT ALIGNMENT

We will refer to three types of texts for a given TV program: *program transcripts*, which are accurately generated transcripts of the program prepared by a human transcriber that do not contain time code and are not prepared in real-time; *closed-captioned text* or simply *captioned text*, which are texts generated by a human captioner that contain a time code for every group of 3-5 words, and may or not be prepared in real-time; and *ASR output*, which denotes text generated by an automatic speech recognition (ASR) system that has a time code associated with each word.

Our goal is to automatically generate closed-caption text by aligning a program transcript with the ASR output. The program transcripts are highly accurate but lack time code information that is necessary to synchronize the text with the speech in a video program. On the other hand, the ASR output contains time code for each word uttered. However, the accuracy of the text produced by current ASR systems is far below the closed-captioned text accuracy. In general, for a wide variety of real-world speech that includes combinations of speech with background noise, degraded acoustics, and non-native speakers, the ASR word error rate varies between 35% and 65%.[4–6] By aligning a program transcript with the ASR output we are able to generate a highly accurate and time-coded transcript, which can then be used as closed-caption text for the video program.

Text alignment is a commonly used technique for machine translation.[7,8] Most of this work has involved the use of parallel texts, where the same content is available in several languages, due to document translation.[9] Given these texts, a first task is to perform a large scale alignment, determining which sentences in the text correspond to which sentences in another language. Once this is achieved, a finer level of alignment may be performed to align the words in sentences. For this application, the two texts belong to different languages and the word pairings that are obtained may be used to derive bilingual dictionaries and terminology databases. A large scale evaluation of text alignment algorithms is available on the Internet.[10] Many researchers have investigated processing texts obtained from ASR and closed-captions for information retrieval purposes. Broadly know as "Topic Detection and Tracking," this area of research aims to develop algorithms for discovering and combining topically related material in streams of data such as newswire and broadcast news in a number of languages.[11] In the context of closed-captions, researchers have used text alignment of ASR text and closed-caption text to train and improve the accuracy of ASR systems.[4]

Given two input text files to be aligned, we first perform preprocessing on the files. Then, using a dynamic programming algorithm we minimize the edit distances between the text sequences to align the words. These steps are described in detail below.

3.1. Preprocessing

The first step of our alignment system is the preprocessing of the input texts. The goal of this step is to convert both texts into a standard form in order to simplify the core alignment procedure.[12] First, we divide the input text into units known as *tokens*, where each token is delimited by white space. Text within parantheses is ignored, since these are generally extra words added by captioners in closed-caption text to provide extra information about the program. Then, each token is processed to remove all punctuation and non-alphanumeric characters. Uppercase characters are also converted to lowercase. For the ASR text, each token is labeled with a time code obtained from the ASR output. Examples of text preprocessing outputs are shown in Table 1.

Original text	After preprocessing
I don't know personally,	i dont know personally
but Mr. Dittemore would,	but mr dittemore would
document S/Agenda/4714, which reads,	document sagenda4714 which reads
quote, "The situation (applause) between	quote the situation between

Table 1. Examples of text preprocessing outputs.

S_1 :	confident	in	a	whirlwind	of	———	change
S_2 :	confident	in	a	world	of	greater	change

Figure 2. An alignment of two text sequences, S_1 and S_2. The first sequence, S_1, contains six tokens and the second, S_2, contains seven tokens. Note that the token "greater" cannot be matched to any token in S_1 so it is matched to a space, represented with a dash.

3.2. Alignment of Token Sequences

After the preprocessing step, the input texts are represented as two ordered sequences of tokens, S_1 and S_2. By alignment of these sequences we mean the following: We find a correspondance between tokens in S_1 and S_2 as to minimize some distance metric. This may require the insertion of spaces, represented with dashes, either into or at the ends of S_1 and S_2, so that every token in either sequence is matched with a token in the other sequence or a space. An example of an alignment of two token sequences is shown in Figure 2.

The objective of aligning a program transcript with ASR output is to determine, for each token in the ASR output, the corresponding token in the program transcript. Once this is achieved, we can determine the time code of the tokens in the program transcript from the corresponding matching token in the ASR output. If no match for a token in the program transcript can be found in the ASR output, then its time code is estimated from the time code of its neighboring tokens.

In order to perform the alignment, a distance metric must be defined between token sequences that measures the quality of a particular alignment, i.e., if the distance is large, then the alignment is poor and vice versa. In the alignment procedure we want to maximize the number of matches, minimize the number of mismatches (such as the "whirlwind" and "world" pair in Figure 2), and token insertions ("greater.") One such distance metric used in sequence matching is the edit distance.[13] The edit distance, $D(i, j)$, for two sequences, S_1 and S_2, is defined as the minimum number of edit operations needed to transform the first i tokens of S_1 into the first j tokens of S_2. The allowed edit operations are insertion, deletion, and substitution. For example, for the sequences shown in Figure 2 we have $D(6, 7) = 2$, since we have one substitution ("whirlwind" and "world") and one insertion ("greater.")

One way to obtain an alignment between sequences is to examine edit distance for all possible alignments of the sequences and choose the alignment with the minimum distance. Unfortunately, this approach is computationally expensive. If S_1 contains M tokens and S_2 contains N tokens, then more than 2^{M+N} comparisons are needed. Even for sequences of just 20 tokens each, this number, $2^{40} \approx 10^{12}$, is large. An efficient method to obtain the total edit distance, $D(M, N)$, between two sequences is to use a dynamic programming method where the problem is divided into sub problems which are solved recursively.[14] In other words, in order to obtain $D(M, N)$ all the possible edit distances are obtained. This is done in an efficient manner using a recurrence relation that is used to obtain $D(i, j)$ based on previously obtained values. Starting from the initial conditions $D(i, 0) = i$ and $D(0, j) = j$, the distance value $D(i, j)$ may be obtained using the recursive the follwoing relationship[13]

$$D(i, j) = \min\left[D(i-1, j) + 1,\, D(i, j-1) + 1,\, t(i, j)\right], \qquad i > 0, j > 0 \tag{1}$$

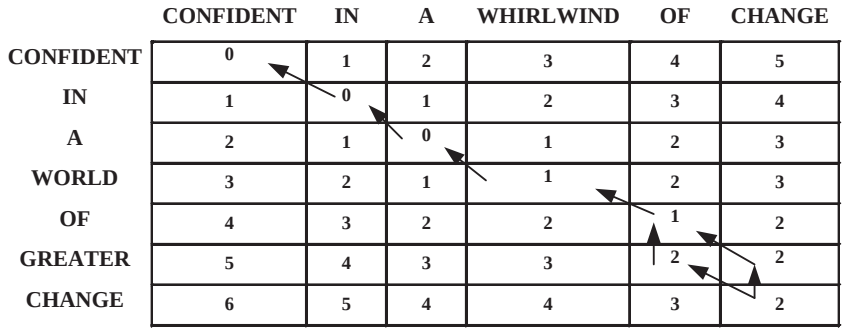

Figure 3. The dynamic programming table for the alignment of the two sequences in Figure 2. The values in the cells are the edit distances and the arrows indicate possible paths.

where the token match function, $t(i, j)$, is defined as

$$t(i,j) = \begin{cases} 1 & \text{if } S_1(i) \neq S_2(j) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $S_1(i)$ is the i^{th} token in sequence S_1 and $S_2(i)$ is similarly defined.

The alignment between sequences can be efficiently obtained by tabulating all edit distances as shown in Figure 3. In this table S_1 appears on the top row and S_2 on the first column. This table is filled as follows: Starting in row one, column one, the subsequence "confident" of S_1 is compared to the subsequence "confident" of S_2; the subsequences are identical so an edit distance value of 0 is entered in $(1, 1)$ in the table. Next, the subsequence "confident" is compared to the subsequence "confident in." Since the token "in" must be inserted into the first subsequence, an edit distance value of 1 is entered in $(1, 2)$. This type of comparison proceeds from left to right across each row until the entire table is filled.

Once the edit distance table is completed a traceback procedure is used to extract the optimal alignment between the two sequences. For each cell in the table, an arrow is placed from that cell to the adjacent cell with the minimum value of the edit distance. Note that, there can be multiple arrows leading away from a cell. These arrows are illustrated in Figure 3. The optimal alignment is then found by tracing back the arrows from cell (M, N) to cell $(1, 1)$. If there is more than one arrow from a cell, the arrow to be followed is chosen randomly; therefore the final alignment may not be unique. The alignment is then recovered from the path by interpreting a horizontal step as an insertion and a vertical step as a deletion. A diagonal step from (i, j) to $(i - 1, j - 1)$ is interpreted as a match if $S_1(i) = S_2(j)$ and as a substitution otherwise.

4. EXPERIMENTS AND RESULTS

4.1. Closed Caption Accuracy

The text alignment algorithm was used to test the real-time captioning accuracy of nine news programs. The accuracy of the captions is determined by first measuring the word error rate (WER), which is defined as

$$\text{WER} = \frac{\text{number of tokens in error}}{\text{number of total tokens in program transcript}} \times 100 \tag{3}$$

The accuracy is then defined as

$$\text{accuracy} = 100 - \text{WER} \tag{4}$$

The accuracies for the nine news programs are shown in Table 2.

Program Type	Meet.	Speech	UN	Brief.	Senate	Speech	Hearing	House	House
Length (minutes)	70	60	40	40	150	30	150	170	140
Words in program	10297	8906	5276	7055	24976	4205	21965	24056	23619
Words in CC	10247	8412	5091	6372	23474	4131	21557	23165	21357
Total correct	9951	7930	4804	5944	22502	4022	20498	21655	19687
Total errors	447	1140	544	1218	2783	227	1874	3068	4390
Misspelled	195	318	215	321	663	65	652	843	1212
Missed	151	658	257	790	1811	118	815	1558	2720
Added	101	164	72	107	309	44	407	667	458
Accuracy	95.66%	87.20%	89.69%	82.74%	88.86%	94.60%	91.47%	87.25%	81.41%

Table 2. The accuracy of closed-caption text generated by captioners for different programs. The abbreviations Meet., Conf, and Brief. refer to a meeting, conference, and a briefing program, respectively

4.1.1. Types of Errors in Closed-Captions

We have classified errors found within the caption text into four broad categories: spelling, interpretation, group, and repetition errors. The captioner causes most of these errors, but some errors may be caused by other reasons.

- *Spelling errors* include words that are missed, added, or misspelled by the captioner. Most added and missed words are due to simple mistakes that a viewer would most likely overlook. For example, some typically missed words are "their" "and" and "the". Some typically added words are "that" and "is". Proper names that captioners are unfamiliar with are the cause of most misspelling errors. Further examples are shown in Table 3.
- *Interpretation errors* occur when the captioner paraphrases the correct word with a similar one. These types of errors are usually caused by acronyms and contractions. Unfortunately, there is no captioning standard that specifies how these types of words should be transcribed. For example, a captioner may choose to transcribe "United States of America" as "USA", "U.S.A.", or "America". Although semantically speaking this example may not be considered a proper error since the two alternative wordings mean the same thing, in our analysis we have counted them as errors. Note that our algorithm may count these cases as multiple errors. For example, due to our alignment algorithm , the term "United States of America" and "USA" would be counted as four errors. For further examples refer to Table 3.
- *Group errors* are blocks of words that are skipped in the caption text or the program transcript. For example, in one of the programs in our data set the closed caption transcript missed a group of 35 words corresponding to 14 seconds of the program. This type of error is most likely caused by intermittent hardware failures.
- *Repeated words or lines* constitute another category of errors. For repeated words a word is simply typed twice by mistake. It is also possible to have several words repeat. Examples of this type of error are given below.
  "such facilities would be unlikely to facilities would be unlikely"
  "peter pace vice chairman of the joint chiefs of vice chairman of the joint chiefs of staff"

4.2. Automatic Speech Recognition Accuracy

The alignment algorithm was also used to test the accuracy of the ASR output. The same methodology that was used in determining closed-captioning accuracy was employed to obtain accuracy for the ASR system. The ASR output obtained from programs were aligned with program transcripts and the accuracies were determined.

	Program transcript	Closed-caption text
Spelling errors	yasin	yassin
	pleuger	ploiga
	salahuddin	london
Interpretation errors	we have	we've
	may 1	may 1st
	it's	it is

Table 3. Examples of different types of errors in the closed-caption texts.

	Program transcript	ASR output
Misinterpretation errors	Dr	Doctor
	4700	four thousand seven hundred
	npc	n.p.c.
Homophone substitution errors	Ba'ath Party	path party
	Hiroshi Yoshisuga	hiroshi you she sued to
	the al Qaeda company	be out candy company
	outside pressure	all side pressure
	they are	there

Table 4. Examples of different types of errors in the automatic speech recognition output.

4.2.1. Types of Errors in the ASR Output

The five types of errors generated by the ASR system are homophone substitution, misinterpretation, group, and program transcript errors, which are described below.

- *Homophone substitution errors* were very common in the ASR output, as expected. Since the ASR system detects phonemes rather than complete words, it is common for ASR systems to generate errors by substituting a correct word with its homophone. This type of error becomes especially prevalent for words that are novel to the ASR system, e.g. proper names. Some examples of this type of error are listed in Table 4.
- *Misinterpretation errors* are similar to the ones described for closed-caption text, which were described in Section 4.1.1. They occur when the program transcript contains a word or number that is different than the ASR output. This happens frequently for numbers. For example, a number was transcribed as "1977" in the program transcript, but was detected as "nineteen seventy-seven" by the ASR system. of this type of error are listed in Table 4.
- *Group errors* happen when the ASR system misses entire blocks of spoken words. For example, the words "al Qaeda were using" were completely missed. In the same program the words "North Korea over nuclear wants" were also missed. The cause of these errors is still under investigation.
- Finally, *program transcript errors* were caused by the errors in the program transcript. For example, the word "al Qaeda" was spelled as "al kindi" in a program transcript.

	Closed-captions	ASR output
Mean	88.06%	65.42%
Median	88.06%	68.42%
Std. deviation	5.1	8.4

Table 5. Comparison of total accuracies for closed-caption text generated by captioners and ASR output.

4.3. Overall Results

The accuracies of closed-caption text and text produced by the ASR system are compared in Table 5. The performance of the ASR system can show large deviations depending on the particular program, audio quality, and speaker. The errors produced by captioners generally reflect the same context as the correct text, albeit with spelling errors. On the other hand, the ASR system can make errors that are totally unrelated to the context of the correct text. For example, the ASR system can detect "Qaeda" as "candy," whereas a captioner can misspell it as "kindi".

5. CONCLUSIONS

In this paper we have presented an algorithm for aligning program transcripts without time code with ASR output that contains time code for each word. Text alignment can be efficiently performed using dynamic programming to find the minimum edit distance between the two texts. We have processed a number of news programs using our system. The error rates for closed-caption texts and ASR outputs for these programs were examined. Finally, we have analyzed the types of errors that occur. Based on these results, we have found that the current state of the art in ASR technology is not sufficient to be used solely to generate closed-captions. The text alignment method that we presented can produce highly accurate closed-captions efficiently.

ACKNOWLEDGMENTS

We would like to thank Dr. Robert Browning of the C-SPAN Archives for his advice and assistance in the project. We would also like to thank Eugene Lin of Purdue University for technical assistance with the alignment algorithm.

REFERENCES

1. G. Robson, *Inside Captioning*. Castro Valley, CA: CyberDawg Publishing, 1997.
2. FCC web site, "http://www.fcc.gov/cgb/consumerfacts/closedcaption.html."
3. K. Jack, *Video Demystified*. San Diego, CA: HighText Interactive Inc, third ed., 2001.
4. M. Witbrock and A. Hauptmann, "Improving acoustic models by watching television," Tech. Rep. CMU-CS-98-110, Carnegie Mellon University, March 1998.
5. J. Choi, D. Hindle, J. Hirschberg, I. Magrin-Chagnolle, C. Nakatani, F. Pereira, A. Singhal, and S. Whittaker, "Scan - speech content based audio navigator: A systems overview," *Proceedings of the International Conference on Spoken Language Processsing (ICSLP 1998)*, 30th November-4th December 1998, Sydney, Australia.
6. S. Srinivasan and D. Petkovic, "Phonetic confusion matrix based spoken document retrieval," *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR 2000)*, July 24-28 2000, Athens, Greece, pp. 81–87.
7. W. A. Gale and K. W. Church, "A program for aligning sentences in bilingual corpora," *Computational Linguistics*, vol. 19, no. 1, pp. 75–90, 1993.
8. I. D. Melamed, "Bitext maps and alignment via pattern recognition," *Computational Linguistics*, vol. 25, no. 1, pp. 107–130, 1999.
9. C. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

10. The ARCADE project web site, "http://www.lpl.univ-aix.fr/projects/arcade/index-en.html."
11. A. Martin and M. Przybocki, "Nist 2003 language recognition evaluation," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech 2003)*, September 1-4 2003, Geneva, Switzerland.
12. D. T. F. Popowich, P. McFetridge and J. Toole, "Machine translation of closed captions," *Machine Translation*, vol. 15, pp. 311–341, 2000.
13. D. Gusfield, *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, New York, NY: HighText Interactive Inc, 1997.
14. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, second ed., 2001.