Fast Caption Alignment for Automatic Indexing of Audio

Allan Knight, University of California, Santa Barbara USA
Kevin Almeroth, University of California, Santa Barbara, USA

Abstract

For large archives of audio media, just as with text archives, indexing is important for allowing quick and accurate searches. Similar to text archives, audio archives can use text for indexing. Generating this text requires using transcripts of the spoken portions of the audio. From them, an alignment can be made that allows users to search for specific content and immediately view the content at the position where the search terms were spoken. Although previous research has addressed this issue, the solutions align the transcripts only in real-time or greater. In this paper, the authors propose AUTOCAP. It is capable of producing accurate audio indexes in faster than real-time for archived audio and in real-time for live audio. In most cases it takes less than one quarter the original duration for archived audio. This paper discusses the architecture and evaluation of the AUTOCAP project as well as two of its applications.

Keywords: Audio Processing; Indexing; Multimedia; Natural Language Processing; Speech Recognition

Fast Caption Alignment for Automatic Indexing of Audio

Over the past 10 years, automatic speech recognition has become faster, more accurate, and speaker independent. One tool that these systems rely on is *forced alignment*, the alignment of text with speech. This application is especially useful in automated captioning systems for video play out. Traditionally, forced alignment's main application was training for automatic speech recognition. By using the text of recognized speech ahead of time, the Speech Recognition System (SRS) can learn how phonemes map to text. However, there exist other uses for forced alignment.

Caption alignment is another application of forced alignment. It is the process of finding the exact time all words in a video are spoken and matching them with the textual captions in a media file. For example, closed captioning systems use aligned text transcripts of audio/video. The result is that when the audio of the media plays, the text of the spoken words is displayed on the screen at the same time. Finding such alignments manually is very time consuming and requires more than the duration of the media itself, i.e., it cannot be performed in real-time. Automatic alignment of captions is possible using the new generation of SRS, which are fast and accurate.

There are several applications that benefit from these aligned captions. Foremost, and quite obviously, are captions for media. Providing consumers of audio and video with textual representations of the spoken parts of the media has many benefits. Other uses are also possible. For example, indexing the audio portion of the media is a useful option. By aligning media with the spoken components, users can find the exact place where text occurs within the audio content. This functionality makes the media searchable.

The technical challenge is how to align the transcript of the spoken words with the media itself. As stated before, manual alignment is possible, but requires a great deal of time. A better solution would be to find algorithms to automatically align captions with the media. There are, however, several challenges to overcome in order to obtain accurate caption timestamps. The first is aligning unrecognized utterances. No modern SRS is 100% perfect, and therefore, any system for caption alignment must deal with this problem. The second challenge is determining what techniques to apply if the text does not exactly match the spoken words of the media. This problem arises if the media creators edit transcripts to remove grammatical errors or other types of extraneous words spoken during the course of the recorded media (e.g., frequent use of the non-word "uh"). The third challenge is to align the caption efficiently. For indexing large archives of media, time is important. Therefore, any solution should balance how much time it takes with the greatest possible accuracy.

The work discussed in this paper is part of a project called AUTOCAP. The goal of this project is to automatically align captured speech with their transcripts while directly addressing the questions above. AUTOCAP includes of two previously available components: a language model toolkit and a speech recognitions system. By combining these components with an alignment algorithm and caption estimator, developed as part of this research, we are able to achieve accurate timestamps in a timely manner. Then, using the longest common subsequence algorithm and local speaking rate, AUTOCAP can quickly and accurately align long media files that include audio (and video) with a written transcript that contains many edits, and therefore, does not exactly match the spoken words in the media file.

While other researchers have previously addressed a similar problem (Hazen, 2006; Moreno & Jeorg, 1998; Placeway & Lafferty, 1996; Robert-Ribes & Mukhtar, 1997), they use different techniques and do not accomplish the task as fast as AUTOCAP can. The cited projects either do more work than is needed, such as a recursive approach (Moreno & Joerg, 1998), or add more features than are needed (Hazen, 2006), for example, correcting the transcripts. In either case, both approaches, while very accurate, take real-time or longer to align each piece of media. And as mentioned previously, for processing large archives of media, shorter processing times are critical. Finally, and most importantly, these works do not address the issue of edited transcripts.

Our research shows that AUTOCAP can accurately and efficiently align edited transcripts. AUTOCAP's accuracy, as measured by how closely aligned the spoken words are with when the text appears on the screen, is well within two seconds of the ground truth. This two second value is what other research cites as the minimum level of accuracy (Hazen, 2006; Moreno & Jeorg, 1998; Robert-Ribes & Mukhtar, 1997). Furthermore, in most cases, AUTOCAP is well below this two-second threshold. Also, it is capable of aligning captions in faster than real-time. That is to say, it can align the transcripts in time no greater than the length of the recorded audio itself. In most cases, it produces accurate alignments in approximately 25% of real-time. This result is possible using a system implemented in Java.

The remainder of this paper is organized as follows. Section 2 provides more details about the challenges of caption alignment. Section 3 describes the AUTOCAP architecture and the tools and algorithms it uses. Section 4 examines our claims about the accuracy and efficiency of AUTOCAP. Section 5 describes in greater detail the previously mentioned related work along with other similar research. Finally, Section 6 provides a brief summary of our findings and final remarks about the AUTOCAP project.

Aligning Captions

Caption alignment is a specialized problem for automatic speech recognition. This section outlines the specific problems that AUTOCAP addresses. It also specifies which problems it does not address. The main functionality of AUTOCAP is forced alignment. As AUTOCAP is not useful for automatic speech recognition training, we start by describing the usual purpose of forced alignment, then differentiate the purpose of AUTOCAP forced alignment, and finally, offer details about the real application of AUTOCAP and how it can be used to enrich media.

The following subsections discuss the major concepts associated with aligning audio media and transcripts. Their purpose is to create a common understanding of the terms used throughout this paper for the sake of clarity.

Forced Alignment

Usually forced alignment is associated with SRS training. By feeding a known collection of utterances to an SRS, it can learn to properly map utterances from audio signals to text. The process involves first breaking the known utterances into individual phonemes and then aligning them with recognized phonemes from the audio source. Modern SRSs uses the Viterbi algorithm for performing these alignments.

Other applications of forced alignment also exist, and not necessarily at the same linguistic level. For example, AUTOCAP aligns audio with written transcripts. For this problem, there is no need to match at the phoneme level (though the SRS will still operate at this level), but instead operates at the word, or even text segment level. Here the goal is not to train the SRS, but rather to align an already transcribed text to an audio file for other purposes than SRS training.

Media and Transcripts

While there are many reasons for alignment of media and transcripts, there are three major reasons we deem important. First is accessibility. Closed captioning has existed for many years. However, in today's media rich world, captioning is a vital part of maintaining accessibility for people of differing capabilities. The problem is, however, that finding the time that each utterance or transcript segment is spoken is time consuming. Automatic means of aligning captions and media provide a more scalable solution for this problem. Such techniques are particularly important as more and more media content are produced.

Indexing is also a powerful tool driven by the growing availability of media and the increasingly varied ways in which it is used. Indexing allows media consumers to search for the exact content that interests them. Since most current indexing technologies require some form of text to associate with the media, alignment of text and audio media is a powerful means of indexing audio media. Other characteristics of media may be used in the future, but the textual content of media will always maintain a basic level of importance for quickly searching media.

Finally, internationalization is also a major concern as the global economy continues to expand and evolve. By aligning textual transcripts with media, content providers not only provide caption and indexing capabilities in the native language of the media, but can also provide translations for multiple languages. This added benefit provides access to a larger audience of consumers for media content.

Edited Transcripts

For the set of media and transcripts on which we tested AUTOCAP, we used edited transcripts. These were transcripts professionally edited by experts with domain-specific knowledge in the fields addressed by the media.

Aligning edited transcripts with media has its own unique set of problems. First, unlike the work by Hazen (2006), the transcripts were considered correct and no additional editing was necessary. However, because the transcripts were edited, they often did not match verbatim what was said in the audio media. This fact imposed two problems for the normal forced alignment problem. First, not every word spoken in the audio was reflected in the transcript. Mistakes by the speaker, such as stuttering or using filler non-words such as "um", were removed from the transcript. Second, not every word in the transcript was necessarily spoken in the audio. For example, if the speaker used the wrong word, the edited transcript instead included the correct phrasing. For these two reasons, aligning the two media at a lower linguistic level is not only a much harder problem but also unnecessary.

Aligning Edited Transcripts with Media

The application for which AUTOCAP is intended is very specific. When content producers wish to take edited transcripts and align them with audio or video content, AUTOCAP can accomplish this task not only accurately, but in faster than real-time. Also, because AUTOCAP allows for edited transcripts, the basic problem is reduced to edit distance, and therefore the longest common subsequence algorithm is used to align audio and text. The following two sections discuss how AUTOCAP accomplishes this task and describes how AUTOCAP is able to perform it accurately.

AUTOCAP

AUTOCAP employs five processing steps that are necessary to align a transcript with its audio. First, the audio file, sometimes as part of a media file that includes video, must be transcoded into a Sphinx compatible codec. Second, a language model is built using the Carnegie-Mellon University Cambridge Statistical Language Modeling Language toolkit (CMU-CAM). Third, both the audio and language model are then used as input to the Sphinx SRS. The SRS produces a list of utterances. Fourth, AUTOCAP aligns these recognized utterances with the transcript and, where unable to use exact timestamps, estimates the timestamp instead. Finally, a

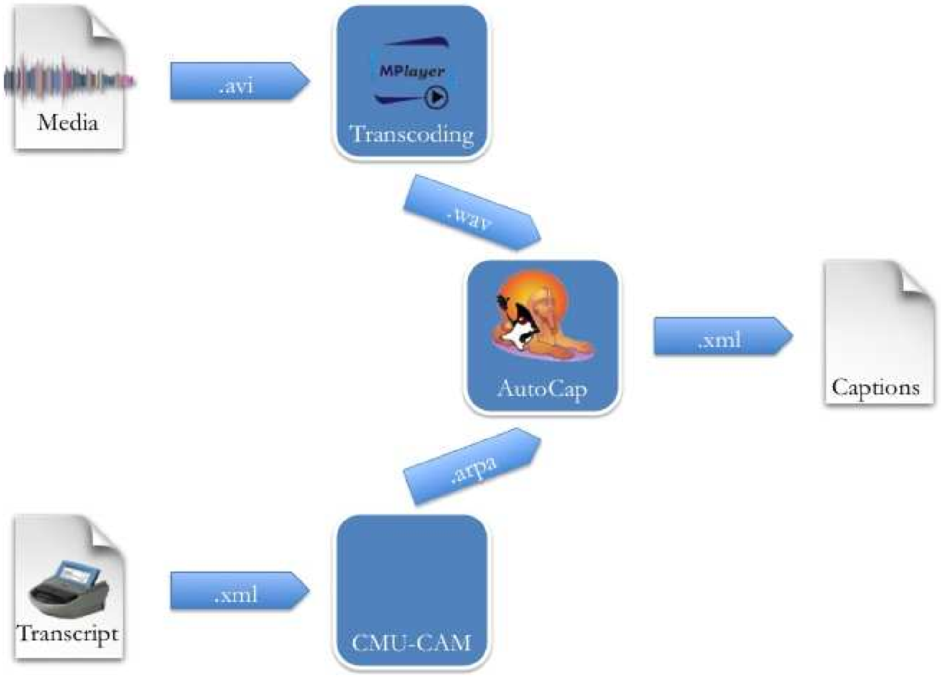


Figure 1. AUTOCAP System Architecture

transcript file is produced that contains all the segments used for captioning and the necessary timestamps to synchronize with the audio/video media.

AUTOCAP is not simply a software program, but rather is a software system integrated with a Java program that performs alignment. The purpose of this section is to describe the entire AUTOCAP system as well as the software itself and how they interact to accomplish caption alignment and audio indexing. Upon reading this section the reader should expect to have a good understanding of how AUTOCAP accomplishes this task. Figure 1 illustrates all of the components that make up the AUTOCAP architecture.

Architecture

The architecture of AUTOCAP is composed of two levels: the system and the software levels. The system level represents the collection of tools, both previously available and those developed as part of this effort, used to perform the task of caption alignment. Figure 1 outlines this level and illustrates the flow of media through the system. The media starts as a file and a transcript file. Once all processing is complete, it outputs the same transcript used as input, with time codes for each transcript segment. The software level represents the actual programming code written as part of this research project by the authors. Its entire contents are original to the project. Figure 2 outlines this level and illustrates the flow of media through it. Figure 2 is an expanded view of the AUTOCAP element shown in the middle of Figure 1. The software takes as its input both the original transcript and the audio portion of the original media file. The transcript is normalized to remove capitalization for alignment later in the process, and the SRS to retrieve as many recognizable utterances as is possible from the audio portion of the original media. As in the system level, the output of this level is the time coded transcript file. The rest of this section describes the various processes used by AUTOCAP to align transcripts and audio media.

Media Transcoding

Before alignment can begin, the media must be converted to an appropriate format. Furthermore, if the media includes videos, the Java Speech API (JSAPI) (Sun Microsystems, 2009) requires that the video be stripped from the media. Once the video is removed, the audio must be encapsulated in a header readable by JSAPI at an appropriate sampling rate and in a suitable codec. To accomplish this task, AutoCap uses MPlayer (The MPlayer Project, 2008). This general-purpose media player can transcode a wide range of audio and video formats as well as change frame rates and sampling rates. Using this freely available open source tool, we found that we were able to convert just about any media file to suit the requirements of the JSAPI.

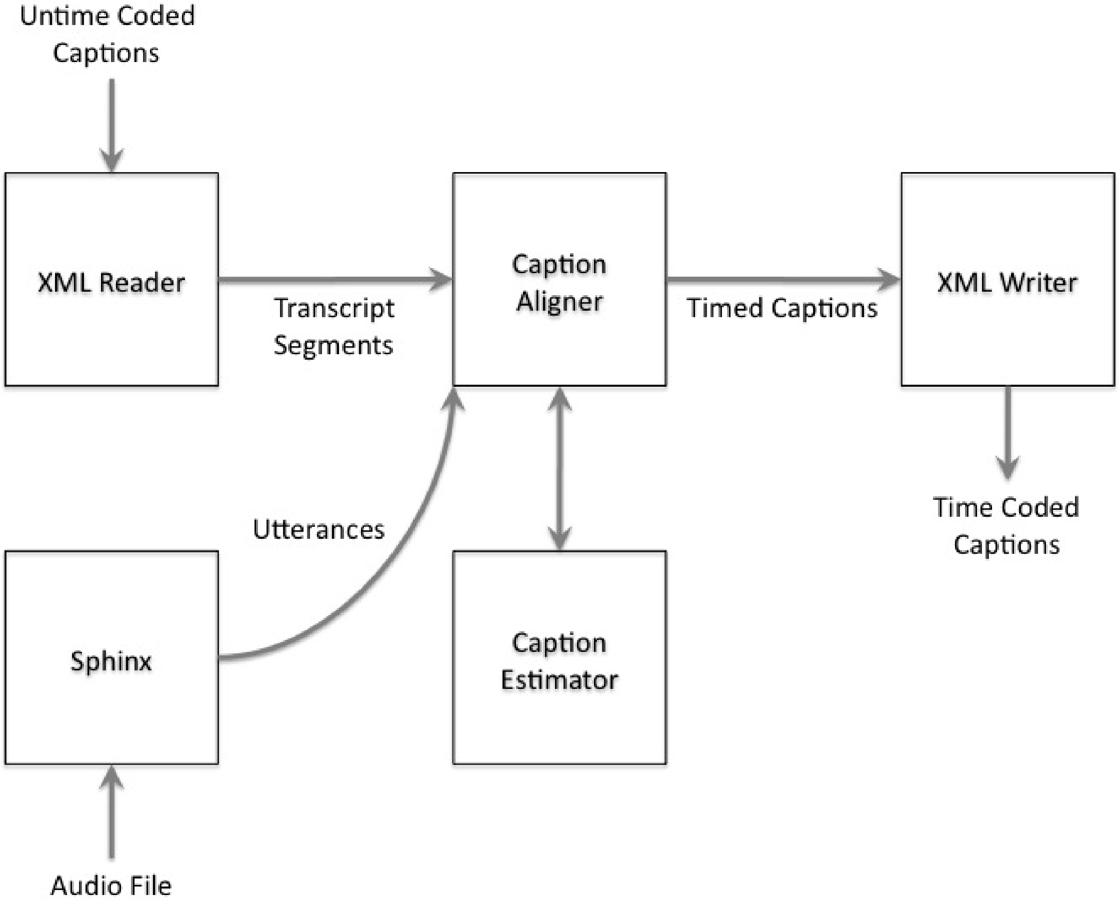


Figure 2. AUTOCAP Software Architecture

Building the Language Model

In order to decrease the word error rate of automatic speech recognition, it is first necessary to create a language model. Since the edited transcript file contains the exact language model of the media, AUTOCAP uses it instead of a larger, static language model. We have observed a reduction in the Word Error Rate (WER) on the order of 25% to 40% by using the transcript to build the language model. For this purpose, AUTOCAP uses the CMU-CAM Statistical Language Modeling Toolkit (Clarkson & Rosenfeld, 1997).

First, the text is stripped from the transcript, removing all XML tags. Then the raw text is fed into a pipeline of tools that create a language model for use with automatic speech recognition. The language model is saved in the Advanced Research Project Agency (ARPA) file format.

Recognizing Speech

Once the media is extracted, transcoded, and the language model is built, the SRS takes caption and media files to begin the process of aligning the captions. The SRS provides two pieces of information necessary for alignment. First, it recognizes as many utterances as it can. Second, it provides timestamps for each of the words recognized in each utterance. Each utterance is made up of consecutive recognized words and is retained for alignment during the next stage of processing.

AUTOCAP uses the Sphinx SRS (Huang & Hon, 1992) from Carnegie-Mellon University. This SRS was selected for several important reasons. Most importantly, Sphinx is open source and provides an intuitive API. Second, because it is implemented in Java, it runs on multiple platforms with no modification. Finally, Sphinx is a speaker-independent SRS. Because the corpus of media we acquired for testing pre-existed, training the SRS would have been impossible. Furthermore the speaker independence feature allows for multiple speakers during a media presentation.

The result of this phase is a collection of utterances. This collection represents a set of anchor points for the alignment phase to match with the transcript during the next phase.

Aligning Speech

The process of aligning utterances with the transcripts is actually a longest common subsequence problem. The application of this algorithm, however, cannot begin until the entire media file has been processed. Using the classic dynamic programming algorithm, AUTOCAP aligns as many words from the transcript as it can while using a minimum burst size. This burst size prevents misalignments, which is especially possible in small utterances of function words, or other common short utterances.

Once the alignment is complete, timings are calculated for each of the segments provided in the transcript. At this point, one of two possibilities occurs. If the first word of the segment was part of a recognized utterance, an exact timestamp for that segment is already available. If, however, the first word is not recognized, an estimation of when the time first word of the segment was spoken must be provided. Providing an estimation of any unrecognized segment start, based on the local speaking rate, is the goal of the next phase of the architecture.

Estimating Captions

At this point in the process, AUTOCAP has recognized as many words as it can and matched those recognized words with the transcript. Within the words, AUTOCAP has indentified "islands" (Huang & Hon, 1992) of recognized words with anchor points at the edges of recognized and unrecognized bursts of words. If the beginning of transcript segments (captions) is within these islands, no more work is required. The timestamp for the word returned by the SRS is used as the timestamp of the caption. If, however, the beginning of the segment is not within an island, then other techniques are necessary to find that timestamp. While Moreno (1998) used a recursive approach to recognize more and more utterances, AUTOCAP uses an estimation scheme that results in similar accuracy and less processing time. Rather than spending more time attempting to do more recognition, it uses two adjacent anchor points and the speaking rate between the two corresponding islands to estimate the timestamp of the first word of a caption.

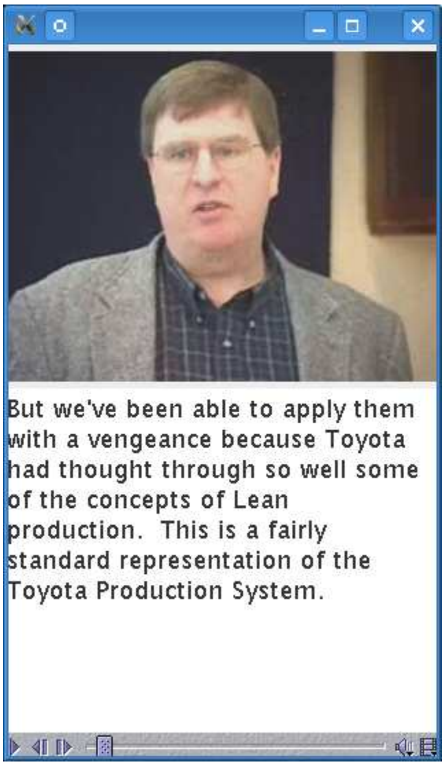
The estimation technique used in AUTOCAP is simple and uses local speaking rate to make estimations. To calculate the estimation of a caption, AUTOCAP counts the number of words between two adjacent anchor points and the difference between their corresponding timestamps. From these two values, a local speaking rate is computed in terms of words per second. Next, it finds the distance from the nearest anchor point to the beginning of the caption. This distance is then multiplied by the speaking rate and added to the closest anchor timestamp to estimate the actual time the first word of a caption is spoken. The formula for this calculation is then:

$$D_{\text{Anchor}_i}(T_{\text{Anchor}_i} - T_{\text{Anchor}_{i+1}})/(\text{Anchor}_{i+1} - \text{Anchor}_i) + T_{\text{Anchor}_{closest}}$$

Outputting Captions

Once all alignments are made, the timestamps are saved, along with the segmented transcripts, producing a caption file. For the files used in developing and testing AutoCap, the original transcript and that produced by AutoCap were the same, except for the timestamps, missing from the original. Other applications of AutoCap need not follow this same pattern.

The resulting caption files are then used to produce a more media rich experience. Figure 3 shows an example of this richer experience. Not only is the video and audio displayed, but captions are as well.



But we've been able to apply them with a vengeance because Toyota had thought through so well some of the concepts of Lean production. This is a fairly standard representation of the Toyota Production System.

Figure 3. Example of using caption files to enhance the richness of a media experience.

Evaluation

Using forced alignment for the purpose of aligning captions is not only possible but also efficient. The following analysis shows that AutoCap is capable of accurately aligning captions using open source technology. Furthermore, AutoCap achieves this alignment using currently available computing hardware in less than real-time. Finally, the transcripts used for the captions need not match word for word with the audio spoken in the media.

Establishing these claims takes several steps. First this document discusses the methodology and equipment used in conducting all of our experiments. Next, it examines the makeup of the experiments themselves and describes collecting all the data used in this analysis. Finally, a discussion of the results and findings of the experiments shows that forced alignment for the purposes of automatic captions is possible using open source tools on commodity PCs.

Methodology

In analyzing the effectiveness of AUTOCAP, a single computer with the following configuration executed all of our experiments: an Intel Core 2 Quad Q6600 running at 2.40 GHz with 2 GB of RAM and using the Fedora Core 8 Linux distribution with kernel version 2.6.23.1-42.fc8. The operating system ran in a typical configuration with X-windows and daemons for SSH and other system functions.

In addition to the hardware and operating system, AUTOCAP and other aspects of the experiments used the following software applications and libraries: AUTOCAP builds with and runs on the standard Java HotSpot Server Virtual Machine build 1.5.0_15-b04 (Sun Microsystems, 2009) and uses the Sphinx4 beta release 1.0 (Carnegie Mellon University, 2004) for its speech recognition engine. For media processing, two utilities were necessary. For language model creation, AUTOCAP used the CMU-Cambridge Statistical Language Modeling toolkit version 2 (Clarkson, 1999) from Carnegie Mellon University. We used this toolkit because it produces language models in the ARPA format and are directly usable by Sphinx. To extract audio and transcode it for use with Java compatible codecs, all experiments used MPlayer version dev-SVN-r26936-4.1.2.

These experiments used all videos from a collection of 26 involving a single speaker with good audio quality. In total, these videos represent 172 minutes of audio, 673 captions and 26,049 words. Altogether, our system spent approximately 501 minutes conducting all of the experiments, excluding the time to transcode and builds languages models.

The source of these videos is a manufacturing consultancy and the content is very domain specific. Experts with the proper domain knowledge edited the produced transcript, which are therefore considered to be completely accurate with respect to their language usage. The experts also created timestamps for the captions manually. We used these manually determined timestamps as the ground truth to compare against our automatically generated timestamps to judge the accuracy of our system. The manually generated timestamps given with the captions, are, however, naturally prone to error. We discuss and quantify this error in the results section.

Experiments

Execution of these experiments involved transcoding each video, creating an appropriate language model for each and then using the Sphinx SRS to align the transcripts. The alignment phase took the bulk of processing time. This process occurred nine times for varying values of the Absolute Beam Width (ABW) parameter used by Sphinx. The ABW directly affects both the amount of work done by the SRS and the accuracy of any recognition. As we discuss this parameter we are using it as a means of gauging the time required to perform the speech recognition phase of the caption alignment and indexing process. We further describe this parameter to give the reader a better idea of how it affects processing time.

As the recognition progresses, the number of possible Viterbi paths increases. Each of these paths represents potential matches for a particular utterance. As the number of paths increases, however, so too does the amount of memory and work required to perform the match. By limiting the number of paths, the SRS can more quickly find possible text matches for the audio. As a consequence of this pruning action, the real match may be pruned, negatively impacting the accuracy of the SRS. The goal, then, is to find a balance between accuracy and the required time for processing. In order to identify the proper balance, the experiments used the following ABW values: 100, 250, 500, 750, 1000, 1250, 2000, and 3000. As the ABW increases, the number of Viterbi paths also increases and, therefore, the amount of processing time also increases, while the WER decreases. The results section discusses the degree to which these parameters are related. For each experiment we saved the caption file, statistics about the resources used, and the accuracy of the experiment.

Results

To discuss the accuracy of the alignments found by AUTOCAP, we require a ground truth. Fortunately, the videos provided to us already included manual caption times. The problem then is how accurate the manual captions are if they are to be used as ground truth.

The media files provided to us contained a video file and a caption file with pre-segmented caption text with timestamps for each. The challenge, then, is to determine the accuracy of each timestamp, specifically, when each segment actually begins. Human determination of these times is precisely the problem, so having another human measure this metric simply adds another source of error. Instead, we used for this study the timestamps from the first word of a caption segment, if recognized by the SRS. These timestamps are accurate to the tenths of seconds, but rounded to the nearest second because the manual timestamps are only accurate to the nearest second. Therefore, to determine the overall accuracy of the manually determined timestamps within the ground truth, we compared all ground truth timestamps to those of the recognized segment starts. The caption error is the absolute value of the distance of the manual timestamps from the actual timestamps as determined by the SRS. Table 1 shows the findings of this phase of the analysis.

Table 1. Results of measuring the accuracy of ground truth.

Total Caption Segments	673
Recognized Caption Segments	408
Percentage of Caption Segments Recognized	60.6%
Total Caption Error in Ground Truth	149 s
Average Caption Error per Caption Segment	0.4 s

The results from Table 1 show that, overall, the manual caption timestamps are within 0.4s of the correct time. For future discussion, we can say that our system is at least as accurate as manual timestamps if they are within the same range. As our later results show, not all alignments achieved this accuracy. However, these automatically generated timestamps led to errors with which people were comfortable. In actuality, people are able to tolerate even longer errors. While we have not found any usability studies that directly address this issue, we believe that caption time stamps within 2 seconds of the actual text being spoken is more than accurate enough.

With regard to the actual accuracy of AutoCap, Figure 4 illustrates the results of the experiments performed. The objectives of the evaluation were threefold. First, our goal was to show that AutoCap exhibited tolerable error rates for caption alignments. Second, accurate alignments should be obtainable in less than real-time. Third, more processing (i.e., higher ABW values) should reduce error rates, but only to a point, beyond which, increased accuracy is minimal and unnecessary. Further discussion of these objectives and the corresponding results follows.

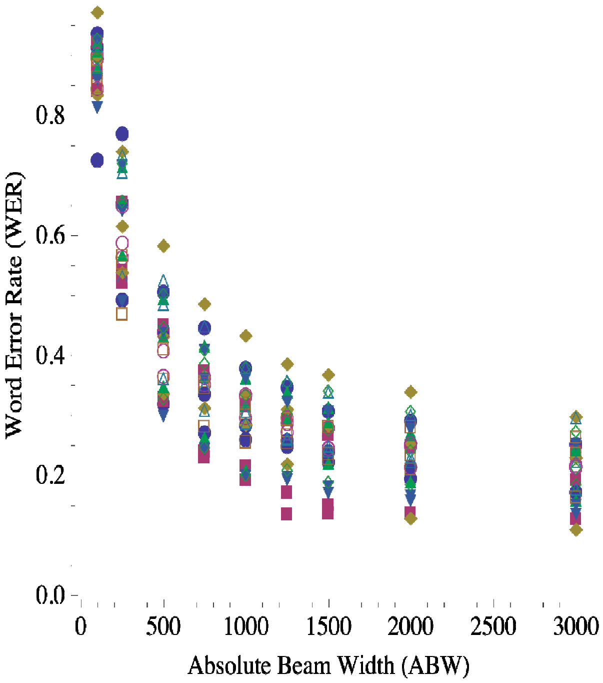


Figure 4. WER vs. ABW

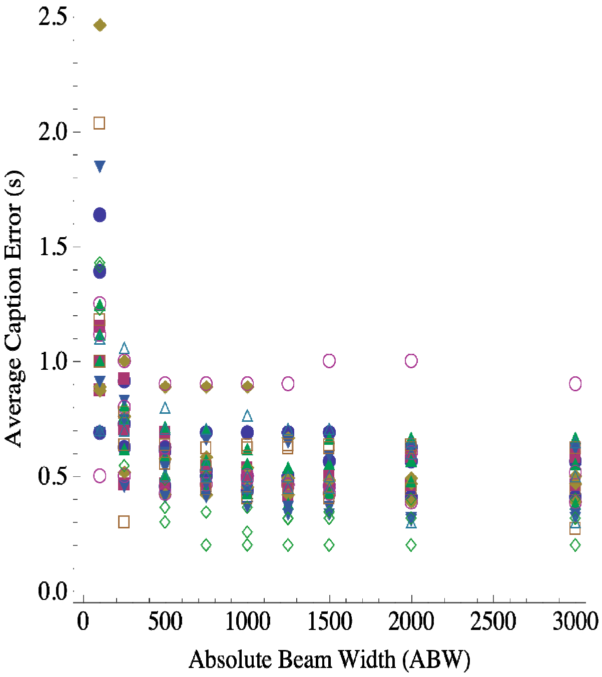


Figure 5. Average Caption Error vs. ABW

The graph in Figure 4 explores the relationship between the WER and the ABW. Along the X-axis are the varying ABW values, from 0 to 3000. The Y-axis records the corresponding WER values and can vary from 0.0 to 1.0. Each symbol represents a different media file. As there are 26 different media files, a complete list is not given. As the ABW increases, the WER decreases. Put another way, as the number of possible utterances tracked increases (and thereby the amount of work for the SRS), the less likely the SRS is to make a mistake. As the SRS makes fewer and fewer mistakes, there should be a corresponding drop in the caption error. Figure 5 verifies this prediction.

Figure 5 is similar to Figure 4. Along the X-axis are the ABW values. Along the Y-axis is the Average Caption Error. We define the average caption error as the absolute value of the timestamp for each caption as found by AUTOCAP minus the timestamp from the ground truth. For clarity, a caption timestamp is for the beginning of a caption segment, not for each individual word. For this graph, the average per media file was calculated and recorded. The results confirm our predictions. As the ABW is increased, there is a corresponding drop in the average caption error rate.

The question still remains, though; about how much more processing can be done in order to further decrease the caption error. Figures 6 and 7 address this question. Figure 6 shows the relationship between the word error rate and the processing time required to align captions. The X-axis records the ratio of processing time to media length time. We use this measure as a means of normalizing the metric over all the media files. The Y-axis records the WER. The individual points also indicate the ABW values used. The graph is similar to the graph from Figure 6, and similar conclusions can be drawn from it. However, there are other trends observable in the graph.

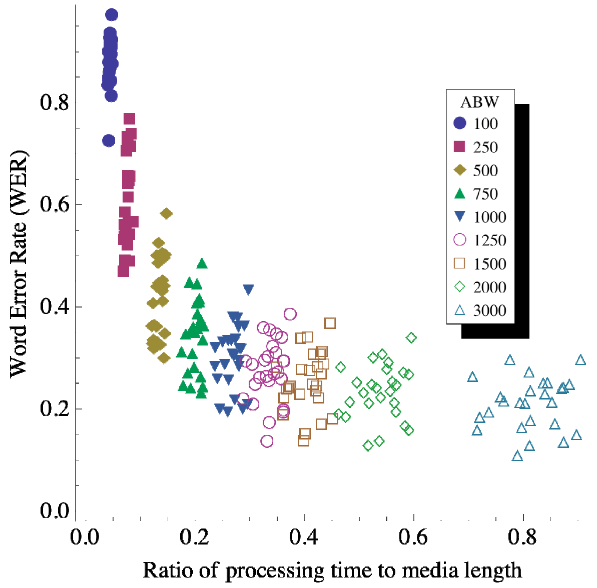


Figure 6. WER vs. Media Length Ratio

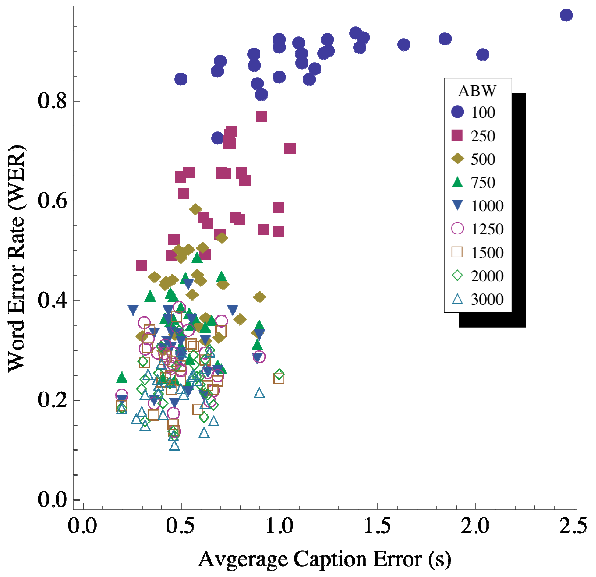


Figure 7. WER vs. Average Caption Error

First, the obvious trend that can be inferred from Figure 6 is that more processing time means lower WER. Second, as the ABW is increased, the variance in the processing ratio also increases. This observation is clear as the clusters of media processed with the same ABW value become more spread out as the ratio tends towards one. Finally, the returns on extra processing time also diminish as more processing time is dedicated to each media file. The main idea behind this graph is that obtaining a WER suitable for caption alignment with AutoCap requires less that real-time. Other systems we have looked at required greater than real-time to align captions.

Lastly, Figure 7 gives more insight into the diminishing return of longer and longer processing times. This graph displays the change in average caption error with changes in the WER. The X-axis records the average caption error and the Y-axis records the WER. While the WER varies from 0.0 to 1.0, the average caption error varies from less than 1.0s to 2.5s. What is observable in this graph is that, as the WER decreases, so too does the average caption error, however, by smaller and smaller amounts. At the bottom of the graph around the point (0.5, 0.2), the results start to bunch up and there is no discernable difference in effectiveness, even with more processing time. This behavior is especially true for ABW values above 1000. A reasonable conclusion is that while more processing time would result in slightly better caption timings, it is not enough to justify further increases in processing time.

Next, we address the question of how much accuracy is needed. As previously stated, for a good experience from a usability standpoint, the captions need only be within two seconds of the ground truth. Therefore, the question is: how much processing is required to achieve the necessary accuracy? The graphs in Figures 8-11 address this question. All four figures represent the histograms of caption errors across all tested media. We varied the ABW to determine an optimal ABW value, and therefore better estimate the amount of time needed to process each media file.

Figure 8 shows the histogram for an ABW value of 100. While the amount of processing required for this setting is about 10% of the length of the original media, the overall distribution of caption errors is not ideal. Caption error for this test ranged from 5 seconds too early to 10 seconds too late. Overall, the number of captions with errors less than or equal to two seconds (the agreed upon threshold) is 83% of the total captions. To achieve more accurate captions, the SRS needs to do more work to increase its overall accuracy, and therefore, generate more captions within the acceptable range.

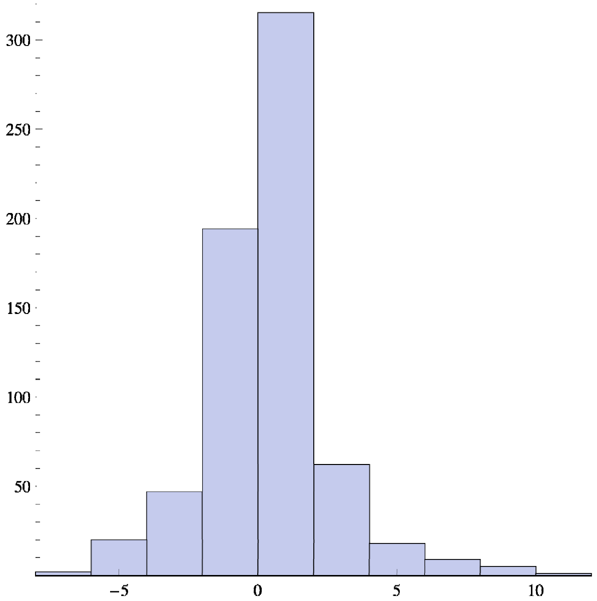


Figure 8. ABW=100

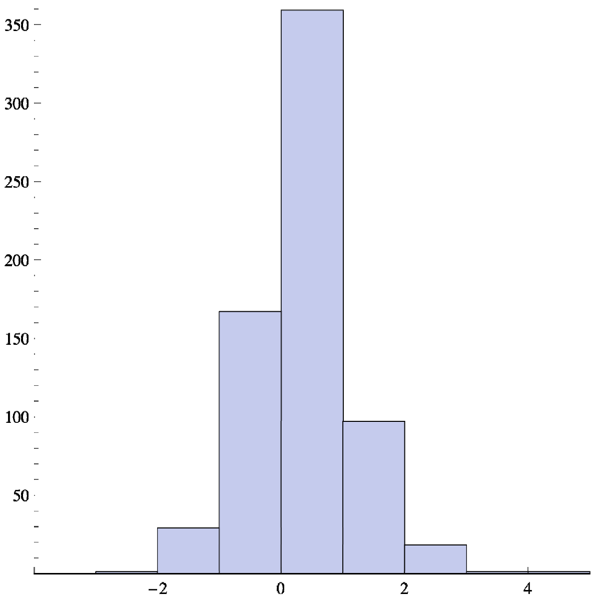


Figure 9. ABW=1000

Here we see that the distribution is closer to the target. First, the number of caption errors within two seconds represents 99.7% of all the captions aligned. Only two captions of the 673 had a caption error greater than two seconds, and no caption errors were greater than four seconds. Also, the number of caption errors less than or equal to one second represents 94% of all captions.

The question still remains, however, of whether more processing leads to even better accuracy. Figures 10 and 11 directly address this question. For these two histograms, the ABW values were 2000 and 3000, respectively. These ABW values represent approximately 2 to 3 times more processing than an ABW value of 1000. Yet, what we see is that they do not yield any more accuracy than with an ABW value of 1000. For each, the number of captions aligned to within the two-second threshold is also 99.7%. Therefore, nothing is really gained, in terms of accuracy, by increasing the processing times by 2 to 3 times. The ABW value of 1000 seems ideal since it requires less than real-time (approximately 25% of the actual media length) to process and still maintains a high accuracy level.

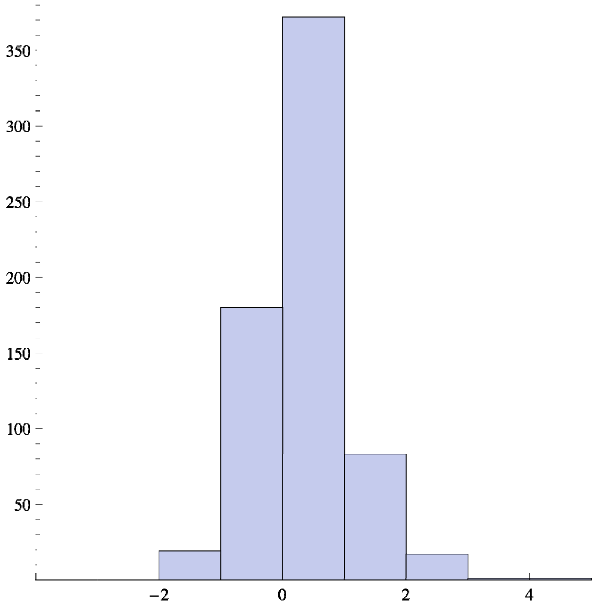


Figure 10. ABW=2000

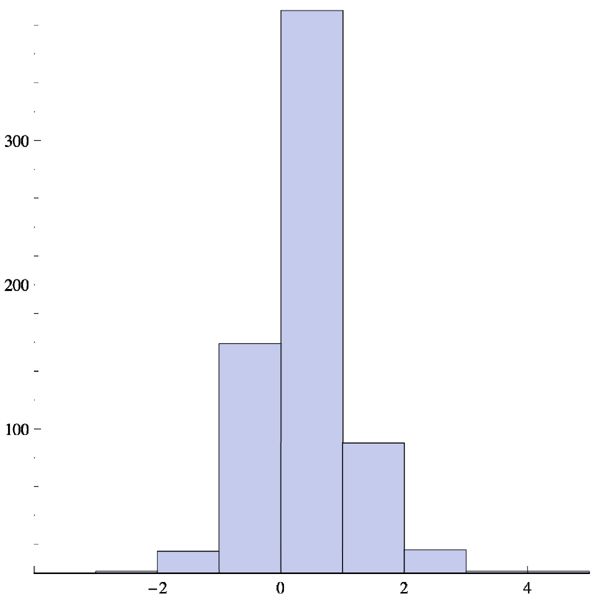


Figure 11. ABW=3000

Table 2 lists all the accuracy results from each of the test runs. The interesting finding is that for this set of media, the ABW value of 1000 is slightly more accurate than for higher ABW values. This anomaly is due either to the inaccuracy of the ground truth (because of human error), or the fact that there may exist more incorrect Viterbi paths with higher scores than for the actual spoken words. Either way, the difference is negligible and unlikely to occur in other collections of media.

Table 2. Processing time and resultant error rates.

ABW	% Time	Error > 1 s	Error >2 s	% Error ≤1 s	% Error ≤ 2 s
100	~5	215	121	68	82
500	~13	50	3	93	99.6
1000	~25	38	2	95	99.7
1250	~33	34	1	95	99.9
1500	~40	35	2	95	99.7
2000	~50	33	3	95	99.6
3000	~80	34	3	95	99.6

Related Work

We have found six projects similar in nature and spirit to AutoCap. First, Moreno suggests a similar technique for transcript alignment (Moreno & Joerg, 1998). However, this work uses recursion to re-process portions unrecognized by automatic speech recognition, and using a more constrained speech domain, recognizes at least part of the unrecognized portions. The recursion process ends once no more words are recognized for all of the unrecognized portions. While this work represents a similar goal to our efforts, it does more work than is necessary for our goals. Aligning pre-segmented transcripts does not require that all words be recognized, just those at the beginning of a segment. Furthermore, our technique gives similar accuracy scores, with in 1 or 2 seconds of the actual the actual time, but with less work overall as ours runs in at most real-time, and usually less.

Second, Hazen (2006) suggest techniques similar to AUTOCAP, but adds correction to the technique. Under this system, it corrects the given transcripts based on processing done by the SRS. Our work specifically deals with edited transcripts from domain experts, and the transcript, therefore, represents the correct textual representation of the spoken words. Therefore, there exists no need for transcript editing. Adding this feature to AUTOCAP would unnecessarily increase processing time and would not lead to more accurate captioning information.

Third, Placeway and Lafferty (1996) use imperfect transcripts, generated in real-time for the purpose of closed captioning, to improve word error rates. The goal of this research, however, is to improve speech recognition, and not to align transcripts and audio for the purposes of captioning. Also, recent advances improve recognition without the necessity of the suggested techniques.

Fourth, Robert-Ribes and Mukhtar (1997) also discusses work that has a similar overall objective to ours. The goal of their project is to hyperlink text to audio recordings. The key here is to find the exact times that the first word of a transcript segment is spoken. Our work similarly finds the beginning of segments, but in a very different way. The major difference between this work and ours is that we use the transcripts themselves to improve recognition. Our system also has the goal of completing a video in real-time or less. Robert-Ribes and Mukhtar ignore the need for real-time processing, which makes processing any large corpus of material overly time consuming.

Finally, two projects use a technique similar to ours for aligning the captions with the audio. Both of the works by Martone et al. (2004) and Huang (2003) use automatic speech recognition and the longest common subsequence to address the alignment problem. Huang uses a slightly different technique: using closed caption (CC) timings to eliminate certain path in their dynamic programming approach. While this technique also works without CC, it loses some accuracy. Martone's work is very similar to ours. This work, however, does not provided much in the way of analysis of the accuracy of the technique. Both projects also do not address the problem of edited transcripts and how they impact accuracy. Nor do they use the transcripts they do have to build language models to increase the accuracy of speech recognition.

Conclusions

In this paper we have described and evaluated AUTOCAP. We have shown that AUTOCAP is capable of accurately and efficiently aligning captions with all sorts of media. Furthermore, with a proper ABW setting that takes into account both speed and accuracy, AUTOCAP can do so in less than real-time and within a tolerable amount of error. And while other projects have similar goals, they do so with more processing than is necessary. While the other works used disparate corpora, making direct comparison impossible, none had the stated goal or conclusion of doing caption alignment in real-time. Also, the related work does not address the issue of edited transcripts. Instead, these other projects expect exact transcriptions of the exact words spoken. Using a tool such as AUTOCAP can lead to more and easier integration of media as well as better and faster indexing of more media types. Using similar techniques as described in this work, researchers and owners of large corpora of media can efficiently and accurately incorporate media into their productions and make them searchable.

References

Carnegie Mellon University. (2004). *Sphinx-4*. Retrieved from

http://cmusphinx.sourceforge.net/sphinx4/

Clarkson, P. (1999). *Statistical Language Modeling Toolkit*. Retrieved from

http://www.speech.cs.cmu.edu/SLM/CMU-Cam_Toolkit_v2.tar.gz

Clarkson, P., & Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge Toolkit. In *Proceedings of the European Conference on Speech Communication and Technology – Eurospeech* (pp. 2707-2710).

Hazen, T. J. (2006). Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Proceedings of the International Conference of the International Speech Communication Association – INTERSPEECH.*

Huang, C. (2003). *Automatic closed caption alignment based on speech recognition transcripts* (Tech. Rep. No. 005). New York, New York: Columbia University.

Huang, X., Alleva, F., Hon, H. W., Hwang, M. Y., Lee, K. F., & Rosenfeld, R. (1992). The SPHINX-II speech recognition system: an overview. *Computer Speech and Language*, *7*(2), 137-148.

Martone, A. F., Taskiran, C. M., & Delp, E. J. (2004). Automated closed-captioning using text alignment. In *Proceedings of the SPIE* (Vol. 5307, pp. 108-116).

Moreno, P. J., Joerg, C., Thong, J. M., & Van Glickman, O. (1998). A recursive algorithm for the forced alignment of very long audio segments. In *Proceedings of the International Conference on Spoken Language Processing*.

The MPlayer Project. (2008). *MPlayer*. Retrieved from

http://www.mplayerhq.hu/design7/news.html

Placeway, P., & Lafferty, J. (1996). Cheating with imperfect transcripts. In *Proceedings of the International Conference on Spoken Language Processing* (pp. 2115-2118).

Robert-Ribes, J., & Mukhtar, R. G. (1997). Automatic generation of hyperlinks between audio and transcript. In *Proceedings of the Conference on Speech Communication and Technology – Eurospeech* (pp. 903-906).

Sun Microsystems. (2009). *Java Speech API*. Retrieved from

http://java.sun.com/products/java-media/speech/

Sun Microsystems. (2009). *Java SE Downloads*. Retrieved from

http://java.sun.com/javase/downloads/index.jsp