Forced Alignment Under Adverse Conditions

Dept. of CIS - Senior Design 2009-2010

Rajarshi Das • Jonathan Izak • Jiahong Yuan • Mark Liberman
{rajdas, izak, myl}@seas.upenn.edu, jiahong@ling.upenn.edu
University of Pennsylvania – Philadelphia, PA

ABSTRACT

The problem of forced alignment is that of matching phonetic segments in an audio sample to its corresponding transcription, which is a vital part of indexing audio files. While various methods have been employed to accomplish this task, the results become less accurate under adverse alignment conditions caused by various disturbances in the audio as well as transcription errors. In fact, the alignment errors are usually left undiscovered until the aligned audio and transcript combination is later reviewed by human eyes and ears, thus defeating the purpose of an automated transcription and alignment process.

This project seeks to develop a robust method to improve existing forced alignment techniques and increase their functionality. This will be accomplished by developing a technique to detect errors in alignment and produce correction algorithms to reduce the frequency of these errors. Various methods to find and fix errors in the alignment process will be examined. By combining these different techniques, a more accurate forced alignment package will be generated, which will be able to operate in adverse conditions found in both the transcript and audio.

1. INTRODUCTION

In today's world, search engines are a crucial part of our lives. Due to technological limitations, however, search engines have, in general, used only text based indexing and searching. In the meantime, we have more multimedia available to us than ever before in both audio and visual form. In order to bring a harmony of the two different trends, it is important to search for some means of converting from one medium to another. In particular, for audio files consisting of speech, it would be natural to expect a text transcription. This could provide a variety of benefits including indexing and searching such data. This process, known as speech recognition, is no easy feat and is constantly being improved.

Speech recognition is essentially the conversion from audio signals of spoken word into text. Speech recognition can also be characterized by a multitude of different parameters, which ultimately relates back to the intended use of the program. The concept of speech recognition was first introduced in 1936 in AT&T Bell Labs. However, due to limits in computing power and practicality, the first official speech recognizers came out in public around the 1950s and 1960s. It took another few decades before the technology hit the commercial market. From the 1980s, speech recognition quickly grew from simple techniques to detect a few hundred words to modern technologies which use much more complicated and advanced learning techniques [2].

Speech recognition today incorporates advanced modeling techniques, various statistical criteria, and state-of-the-art algorithms. The general procedure to attack this problem involves using some controlled training data to generate one or more models. Then the test data is run through these models to search for the "best fit". The term "best fit" must be used loosely in such cases because there are many ways to determine what would best provide such a solution.

Naturally, there is a large amount of linguistic processing involved as well. First, the audio has to be broken into phonemes, the smallest unique segment of sound. However, such a sound does not give a direct correlation to text but instead must be examined based on general location and proximity of other phonemes. The distribution of the phonemes then plays a large part in the next section of analysis which involves modeling the data.

Current techniques use Hidden Markov Models (HMM), which have the benefit of being able to use a sequence of phonemes rather than just each one individually. This does require a sufficient amount of training data though. In addition, these HMMs are combined with a plethora of other algorithms and analysis tools. In our project we look to build upon the Hidden Markov Model Tooklit (HTK), which is open source and used primarily to develop HMMs for speech recognition.

This basic speech recognition tool can be used for a wide variety of purposes, although with many caveats. The goal is to examine a procedure known as forced alignment, which is essentially an optimal alignment of some input transcript with its corresponding audio. If the transcript does not contain errors and the audio consists of clean speech without background noise, very good results can be expected with current software in aligning the two. However, the real issue is when the given transcript has missing words, wrong words, or other worse discrepancies. Then, the issue of alignment becomes much more complicated because the speech recognition also has to be able to detect errors in the given transcript. Furthermore, there can be numerous difficulties in the audio itself. This includes background noise, background music, various pauses in speech, and virtually any other disturbances that detract from the purity of plain speech. The current methods that produce the best results in such environments focus on audio indexing, lacking the precision and phoneme alignment that would be useful in situations that require more meticulousness, such as speech research.

There are a number of circumstances in which aligning an audio source with a preexisting transcript is advantageous over creating a new transcript from the audio using voice recognition. Many words in the English language can be acoustically similar to each other which would force a speech recognizer to decide between multiple words for the same sound. Even with current language models, the wrong word will sometimes be chosen. Human transcription can often avoid these errors as they work from a more suitable dictionary, especially when the individual doing the transcription has knowledge of what is being said. Additionally, humans can transcribe speech accurately in many noisy environments in which current speech recognition techniques perform poorly. In these instances, a more accurate alignment can be produced by aligning the accurate, man-made transcript with the audio source than by using speech recognition to produce the transcript for the alignment.

The Penn Phonetics Lab Forced Aligner Toolkit will be extended in order to make it more accurate under these adverse conditions. This toolkit is more accurate than a number of other acoustic aligners, generally producing alignments with word boundary mean distance errors of less than 50ms on relatively clean audio with reasonably good transcripts. In fact, many of the word alignments are "perfect" [7].

Thus, the goal consists of multiple steps. First, errors will be detected in the given alignment. Next, the reasons behind these errors will be thoroughly examined. This depends on the origin of the transcript. In the case that the transcript itself was made with some other speech recognition tool, there may be patterns in the errors that can be searched for. Here, potential trouble spots include problem words or phrases, or poor choice of statistical ratios. Finally, the last step would be to actually correct the errors using an improved and more robust algorithm. Each of these steps will be delved deeper into as the possibilities using the speech recognition tool are explored.

2. RELATED WORK

2.1 Forced Alignment

With the growing plethora of multimedia content available over the internet, the ability to search audio-visual content is becoming increasingly useful. Searching such content requires time-aligned transcriptions [1]. A number of methods exist for automatically producing such an alignment from a transcript of an audio source for different types of audio conditions. Given the transcript of an audio source where errors in the transcription are unlikely and the audio consists of only clear speech, a single pass approach can be used, such as the method implemented for spoken books, by Caseiro *et al.* Such an approach is not suitable for more common audio conditions with natural disturbances as these would cause errors that could not be corrected.

A recursive approach was developed by Moreno *et al.* that works well under more diverse audio conditions, such as noisy speech signals, even when there are errors present in the provided transcripts [5]. The algorithm runs a speech recognition system using a dictionary and language model produced from the transcript and the resultant hypothesis string is aligned with the transcript. The longest sequences of consecutive words aligned between the transcript and the hypothesis string are chosen as anchors. The anchors are then used to partition the audio and the transcript into aligned and unaligned segments, where the aligned segments are the anchors and the unaligned segments are the regions between the anchors. The algorithm recursively goes through each unaligned segment, redefining the dictionary and language model from the list of words found in the transcript segment that corresponds to the audio segment. The algorithm iterates on each unaligned segment until there are no words left in the transcript segment, the duration of the segment is smaller than a set size, or there is no speech recognized in the segment. The algorithm only selects sequences of a certain length, which decreases dynamically as the algorithm progresses, to be anchors. Larger word sequences have a greater confidence of being correct, thus the algorithm aligns segments that are more likely to be wrong after segments with higher confidence score have been aligned, reducing their impact on the rest of the alignment. Sections of the audio with noisy conditions and errors are likely to have smaller anchor sequences, delaying their alignment to later iterations that employ more restricted dictionaries and language models and have smaller segment durations. This makes alignment easier for the regions that are harder to align and restricts the errors to smaller regions, limiting their negative impact on the alignment to those regions.

On an experimental audio file that had a relatively large percentage (44%) of clean segments, the algorithm correctly aligned 98.5% of words with 0.5 second accuracy. Further experiments showed accuracy was significantly reduced in audio signals contaminated with white noise, increasing the mean of the time error to 2.4 seconds with a standard deviation of 19.4 seconds and reducing the percentage of words accurate with less than 2 seconds to 94.3%. The technique presented by Moreno is aimed at producing results precise enough for indexing audio, which was estimated to be a word accuracy of less than two seconds.

While automated transcription techniques that are highly accurate under ideal conditions have recently become available, there is existing research which focuses on aligning manual transcriptions and other approximate transcriptions. One research piece on the subject consists of an analysis of such transcriptions and a proposed new alignment approach for them that attempts to discover and correct errors in the manual transcription. Analysis of the sample transcriptions revealed an average error rate of 10%. Of these errors, 66% were due to deleted words, or words which were present in the audio source but not in the transcript, and 24% of the errors were due to words in the audio being substituted with incorrect words in the transcript [1]. The paper introduces an alignment method that uses a quick approximate speech recognizer to produce a transcript for the audio which is less accurate than the original transcript and finds anchor points by matching the words in the original transcript to the new transcript. A "pseudo-forced alignment", which is an alignment that allows for deletion of words, insertion of words that appear to be missing which were found by the speech recognizer, and the substitution of phonetically similar words, is produced over the segments between the anchor points. The adjustments made during the alignment process are then applied to the original transcript. The error rate in the transcript was reduced by 12%, mainly through the reinsertion of missing words, and the alignment error rate was 3%. Other types of errors found in the transcripts were rarely corrected.

2.2 Speech Recognition With Noise

Delving further into current technologies, it is clear that a lot of the problem lies in the quality of the audio itself. Much of the data used in various experiments is hand-picked to be clean and well-enunciated. But, the authors always acknowledge that a potential weakness, and often a critical disadvantage, is the possible disturbances in a speech. To maintain generality, we need to understand the problem of speech contaminated with various background noises.

One of the earlier methods to tackle such a problem was to use Hidden Markov Models in signal decomposition as proposed by Varga and Moore [6]. This technique begins with the signal already fully composed and therefore ignores any pre-processing solutions that might be available at an earlier stage. However, HMM decomposition also has many advantages since it can model various changing signals and thus deal with sudden noises as well as more subtle but persistent background noises. Since the signal consists of various component signals that have been combined together, each component has to be accounted for in its own HMM. When running the Viterbi algorithm to find the most likely sequence, the combination of the various HMMs must be accounted for. This modified algorithm was then compared to the baseline technique as well as to another algorithm known as the Klatt Masking Technique. In Klatt Masking, a certain noise mask is determined based on the overall data. This is deemed the threshold and any noises that fall below it are then treated separately. In this paper, such noises were replaced with the mask itself. The speech data consisted of isolated digits, which were superimposed with either pink noise or machine-gun noise. In the results, the decomposition method always performed much more successfully than the other two methods based on the number of words not recognized by each. However, the results have to be taken with some caution as this paper just scratched the surface of the topic. For one, they only dealt with one fixed background noise at once. Furthermore, an important point to note is that both background noises were still very systematically added. A much more robust test would include actual speech in a loud background environment so that the results cannot appear at all to be contrived. Thus, we should add this technique to our arsenal, but put it through a much more rigorous set of tests.

Another technique studied by Urbanowicz and Kantz involved using nonlinear methods to reduce noise by examining the signal beyond second-order statistics [3]. The nonlinear method is compared with a linear method along with a hybrid that switches between the two. Their final results are actually quite scattered depending on the specific nuances of the audio file, such as frequency and amount of noise. Overall, they do claim some success in improving the audio file for a specific commercial speech recognizer. The big caveat, though, is that the techniques used introduced new distortion of its own. In some cases, it did improve recognition capabilities, but they were unsure how it would hold up for various audio. They recognized that further work could be done in expanding the functionality to a more diverse set of sounds.

One particular technique proposed classifying audio in five distinct classes: silence, music, background noise, pure speech, and nonpure speech. Then, by determining the boundaries between the different sections, it would be possible to handle each case by itself. Lu *et al.* discovered that using multiple support vector machines, they were able to segment the audio to a high degree of accuracy [4]. While this may be beneficial as a stepping point for certain other techniques, it does not actually clear up the nonpure speech.

The limitation that appears repeatedly in the noise reduction experiments is that each technique seems to work in a specific case, but not necessarily in a broad set of cases. Thus, there are ways of improving the system. It is also interesting to note, that some research in the past has attempted to model the noise, while other research has attempted to get rid of noise altogether. In order to achieve the goal of improving forced alignment, both of these novel ideas, as well as a hybrid of existing techniques, will be examined.

2.3 Support Vector Machine

Support vector machines (SVMs) are a set of learning methods that may be used for regression. Regression is the analysis of the interaction between a set of dependent and independent variables. A support vector machine constructs a hyperplane that separates two different classes of data points. Since a perfect separation may be impossible, a hyperplane that produces the cleanest separation of the two classes of datapoints is used. The best separation is then the hyperplane with the largest distance from any datapoint since this will lead to the least error when classifying new datapoints using the hyperplane. Studies have shown SVM implementations to perform well in regression compared with other methods.

3. SYSTEM MODEL

The project will consist of a multi-faceted approach to improve the current forced alignment software. In the beginning, the main focus has been to approach this problem when the transcript has certain issues. First, errors in the forced alignment have been be identified. Next, the errors have been classified into certain categories based on each of their causes so that all errors in one category are treated in a similar fashion. It is anticipated that certain problem areas will include missing text, incorrect text, or misalignment due to the audio not having continuous speech. Then attempts will be made to correct for each type of error and improve the current forced algorithm procedure. Another key focus of the research will be to split the audio file if it is not continuous speech so that intermittent noises between speech do not detract from the alignment algorithm. This will be extremely beneficial in audio indexing which is vital to search engines. Furthermore, it is likely some of the techniques used can be also be put to use in a general speech recognition algorithm.

Even though this is the main focus of the project, there is another whole aspect that can be examined after the completion of the above endeavor. Initially, special attention was given to the transcript. Similarly, audio files will be examined to see what kind of improvements can be made to the signals to facilitate this process. This problem opens up a whole slew of other issues to examine, particularly how to isolate the speech in audio from all possible background noises, including other speech. The techniques already available will be examined rigorously, and attempts will be made to generate some hybrid method to help solve this challenging problem. This portion of the project is much more ambiguous due to the various challenges we expect to face. However, a solution to such a problem can hopefully increase success rates of available commercial speech recognition tools.

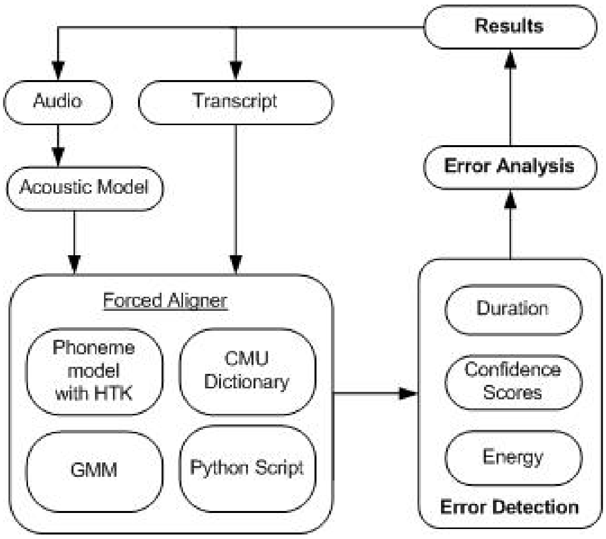

Figure 1: Proposed Approach

The implementation of these tools expand upon the Penn Phonetics Lab Forced Aligner (P2FA), an automated phonetic aligner based in HTK. It consists of an acoustic model that is made up of Gaussian Mixture Model (GMM) based monophone HMMs on 39 perceptual linear predictive (PLP) components. The process begins by analyzing the audio sample using the acoustic model in order to extract the relevant linguistic information. The signal is analyzed using a sliding window method such that one frame, or window, of analysis consists of a fixed duration. The analysis begins at the beginning of the signal and the data extracted from the current frame is represented as 39 PLP coefficients. The next segment of signal to be analyzed begins at a fixed time after the last segment began, which may be overlapping, and is represented in the same manner. This process is repeated until the end of the audio signal is reached. The transition from one segment to the next is analogous to a window sliding from one segment of the signal to the next such that, in each iteration, the window slides the same amount of time and contains a segment of the same fixed duration. The forced aligner uses speech recognition to produce a forced alignment with a phoneme model, a phonetic dictionary, and the transcript of the audio file. The phoneme model has been trained using HTK over large data sets obtained from the Linguistic Data Consortium (LDC) and the dictionary used was the Carnegie Mellon Unversity Pronouncing Dictionary. Multiple HMMs are used to model the speech and align it with the transcript. Each model produces confidence scores for each word from the GMM of each state in that model based on the PLP coefficients that represent each audio segment. A confidence score is calculated for each HMM and the one producing the greatest score is used for the alignment. These existing components are represented in Figure 1 by the components without bold text.

As discussed above, the plan to improve forced alignment in adverse conditions is three fold: error detection, error analysis, and error correction. The bold components of Figure 1 represent these three areas. To detect errors, the confidence scores, in addition to other statistical criteria, produced by the aligner are being analyzed. This information is crucial in revealing the distribution of phone boundaries produced by the different HMMs along with their individual confidence scores. It is expected that alignment errors will originate from the audio file, the transcript, or the actual forced alignment procedure. At first, mainly the two latter sources will be examined. The first method will be to use a duration based analysis of our forced alignment. Observing the behavior of the aligner has demonstrated that errors in the transcript produce temporal anomalies in the alignment process. For words and phones with high confidence scores, the approximate tempo of the speech and duration of the different phonemes can be determined. Using this information, an expected range of the duration of other words and phonemes can then be calculated. This allows errors to be discovered by comparing the duration of phonemes and words in our alignment with this expected range. Speech recognition techniques may also be combined with a semantic language model implemented as HMMs to determine the presence of missing, added, or incorrect words.

The precise implementation of error correction cannot be specified without further experimentation with the existing aligner. As of now, it has been determined that one definite improvement will be to mark boundaries on the audio file to differentiate areas of speech and non-speech. Furthermore, other algorithms will be studied and changed to make them more robust so they work when a full or correct transcript is not available. Some of the ideas to detect errors can be extended to also fix these errors, such as the duration and semantic models that will be explored.

The experimentation, training, and test data required is from the University of Pennsylvania's LDC catalog of corpora. This includes some of the popular corpora that have been used by other research groups in the past. The audio data currently being employed is the cleanest data possible but as the alignment technique improves, less stringent data will be required. Since this project is attempting to improve forced alignment algorithms under specific audio conditions, we have modified some of the audio data obtained by adding noise and other audio to the signals to create training data and a gold-standard data set for evaluation purposes.

4. SYSTEM IMPLEMENTATION AND PERFORMANCE

In order to discover the behavior of the P2FA aligner when a transcript that contains errors is given as input, a large amount of transcripts that contain errors to analyze are required. Therefore, a module was designed and implemented that takes a transcript and produces numerous similar transcripts with added errors. There are three types of errors that the module creates in the transcripts files: deletion errors, insertions errors, and replacements errors. Deletion errors are produced by removing one or more words from the transcript at random. Insertion errors are produced by inserting one or more random words from the dictionary used by the aligner into the transcript at random locations. Replacement errors are produced by replacing one or more words in the transcript, chosen at random, with random words from the aligner's dictionary. These transcripts are then aligned to their corresponding audio files using the

P2FA aligner.

For further examination of the forced alignment errors in the current system, the results from using the P2FA were juxtaposed with predetermined forced alignments found in test data obtained from the LDC. The data set of choice was the TIMIT Acoustic-Phonetic Continuous Speech Corpus. TIMIT contains recordings as well as pre-aligned phonetic transcriptions of 630 different speakers, each of whom read ten different sentences. All the TIMIT corpus transcriptions have been hand-verified and thus provide a solid benchmark for comparison with the aligner. To that end, a Python Script has been generated that is able to perform the alignment in batches and can generate multiple Praat text grids. Praat is an acoustic analysis program used to examine the final results of the aligner using text grids: files that contains the timing of each phoneme in a transcript. Therefore, each of these files essentially contain the forced alignment for a specific wave sound file and the various errors our module interjected in the transcript. Using a python script, the P2FA alignments and the TIMIT data set alignments have been compared to determine the differences in timing.

All the scripts have been implemented using Python. Python is being used for the programming because of its readability. Additionally, the P2FA that is being expanded upon was written in Python. Maintaining the same programming language will make it easier for others to further expand upon the aligner in the future.

There are currently two methods being examined that will improve the detection of alignment errors. Each of these methods are being implemented separately, in order to determine the results of each method independently, and will later be used in combination in the final error detection tool. The first method uses temporal analysis to compare the duration of the segment each phone is aligned to with the expected duration of each phone. A distribution of the expected length of each phone will be obtained using a script that determines the distributions found in test data.

This distribution is used by the temporal anomaly detection module, which traverses each phone in an alignment and detects anomalies in the duration of segments aligned to phones. The module iterates through each phone in a Praat text grid and examines the duration of each phone, which is then compared with the expected duration of that type of phone. The expected duration of a phone is described by its maximum and minimum duration based on a normal distribution obtained from the test data. This tool is expected to be valuable in finding all types of alignment errors. Testing has shown that errors in the transcript and noise in the audio cause such temporal anomalies to occur as the segments that are misaligned tend to be aligned to minute segments at the boundaries of other words or to large segments of speech and noise that are not accounted for by the phones in the transcript. Since the duration of a phone will vary due to factors such as different speakers and when surrounded by different phones, only large anomalies will be considered significant in error detection. Many transcription errors, such as deletion and insertion, and noise in the audio cause temporal anomalies that vary significantly from the phone durations found in normal speech variations.

Some examples of these temporal anomalies caused by errors are depicted below. Figure 2 show the alignment of a speech sample consisting of the phrase “I eat cheese,” and its proper transcript file. Figure 3 shows the result of alignment when the word “eat” is removed from the transcript, producing a deletion error. The highlighted segment corresponds to the alignment of the word “I,” which consists of only on phone, AY1. The phone is aligned to a large segment of speech that includes the three phones which make up the words “I eat,” as in the correct alignment. The duration of this segment is significantly larger than the expected duration of the phone AY1. Figure 4 shows the results of alignment when the words “do not” are added to the transcript, producing the phrase “I do not eat cheese.” The highlighted segment corresponds to the segment aligned to “do not,” which consists of five phones. These five phones are aligned to the segment of speech between “I” and “eat,” causing the duration aligned to each phone to be significantly smaller than the expected duration of these phones. These two errors would both be detected by the module.

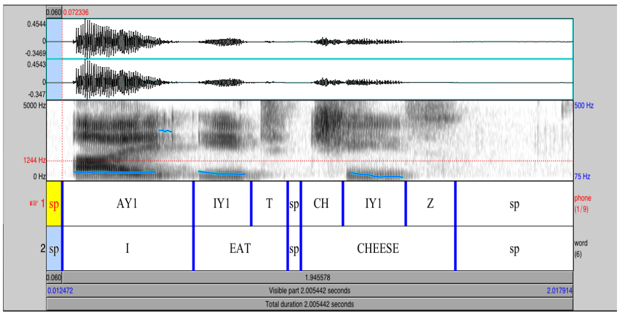
sp
AY1
IY1
T
sp
CH
IY1
Z
sp
I
EAT
CHEESE
Visible part 2.005442 seconds
Total duration 2.005442 seconds

Figure 2: No errors

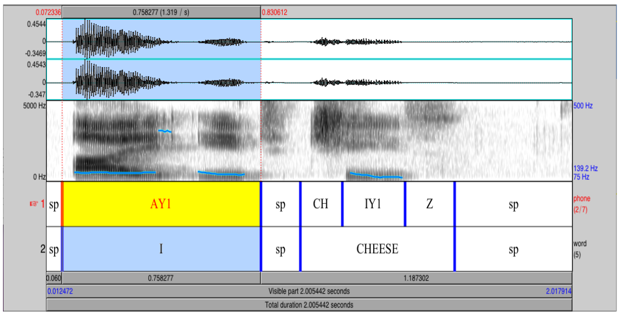
sp
AY1
sp
CH
IY1
Z
sp
I
CHEESE
Visible part 2.005442 seconds
Total duration 2.005442 seconds

Figure 3: Deletion errors

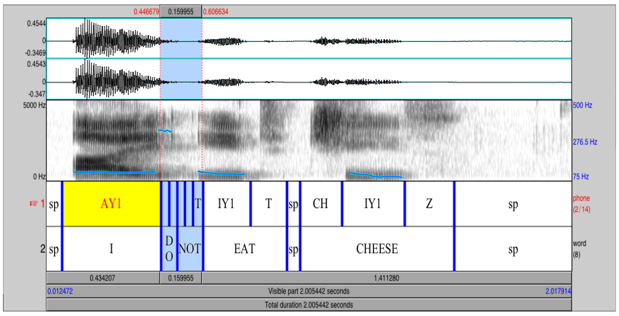
sp
AY1
T
IY1
T
sp
CH
IY1
Z
sp
I
D
O
NOT
EAT
CHEESE
Visible part 2.005442 seconds
Total duration 2.005442 seconds

Figure 4: Insertion errors

If the three categories of transcript errors are more closely examined, the following observations can be made. First, the deletion errors cause the aligner to combine the missing word in the audio with an adjacent word. In the case of insertion errors, the extra words are dealt with a little differently. Sometimes they are aligned with the silences in audio and other times they are aligned with parts of adjacent words. In both these cases, the duration based algorithm is expected to give greater insight on the relative probabilities of every phone being a certain length.

This method can not be relied upon for replacement errors, as these errors are not as likely to produce anomalies in the duration of phone alignments. Due to the nature of HMMs, these errors are fortunately not percolated through the speech recognition model. Instead, the aligner lines up the incorrect word with the correct word it had replaced. Thus, in this step it is more crucial to examine the confidence scores that can be extracted from the model. By extracting these values, we can generate a visual chart to depict the the confidence of our HMM throughout the entire alignment process. Any dips in the data may indicate that the aligner had trouble for those phones. Depending on the actual value of the confidence scores, we then intend to write a script to follow heuristic and mark these areas. This could result in marking the replacement errors.

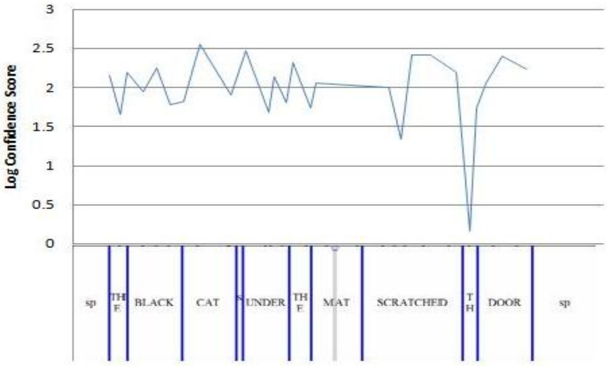

Figure 5: Correct alignment

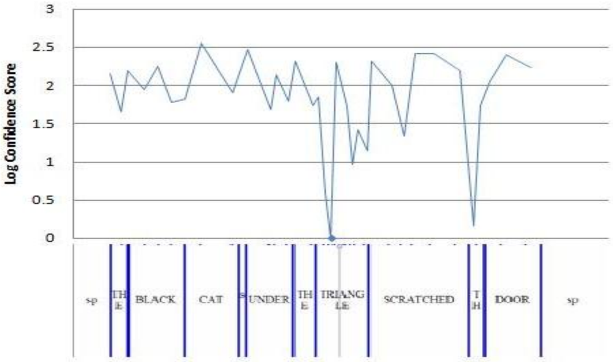

Figure 6: Incorrect alignment

The figures above demonstrate one naive heuristic that could be used to determine the feasibility of the alignment based on the confidence score of each phone. To visualize such dips, the confidence scores outputted by the HTK were first standardized. Since silences generally seemed to produce negative confidence scores, they were removed. Then with a simple heuristic all negative scores were converted to one, so that we could take the logarithm of all the scores. Next the scores were graphed and the outliers were highlighted to find potential problem spots in the alignment.

Figure 5 shows the correct alignment; there are no highlighted outliers as would be expected since the transcript exactly matches the audio. In figure 6, the word “mat” was replaced with the word “triangle”, to show a sharp contrast. The trouble spot is the highlighted point which can be found inside the alignment of the word “triangle”. It is important to note, that this is simply a naive beginning heuristic and may not always hold. However, it does pave the way for further analysis of replacement errors.

The initial examination of both the temporal and confidence score analysis showed that there is potential for building further heuristics based on these models. However, the naïve estimates used above need to be refined to develop a more accurate model.

Data was gathered for each phone from the TIMIT corpus in order to perform further analysis on the duration of phones in speech. This data was then examined and tested using different distributions to determine which type of distribution gave the best representation of the phone durations observed in our experimental data. Unfortunately, most regular distributions failed to provide an accurate model for the data. However, the range of durations for each type of phone proved to be useful information. Durations tended to fall within a certain range and there were very few outliers. Using this information, it was hypothesized that durations in alignment that fall outside of these ranges would indicate a higher chance of an alignment error.

Two distinct methods were implemented to detect temporal anomalies based on this hypothesis. The first method involved determining if each phone duration was an outlier of the range of durations for that phone found in our research data. It was suspected that this method would perform well in the detection of insertion and deletion errors as alignments with these types of transcript errors tend to cause this type of temporal anomalies.

The second method was implemented based on the observation that transcript errors tend to produce misalignments of multiple phones in succession or of multiple phones within the same word. Therefore, anomalies in the duration of multiple phones in succession or within the same word was hypothesized to be indicative of an error. Such anomalies would include multiple phones in a word or in succession that are within the fringe their respective standard duration range.

There were a number of approaches that were considered in the implementation of this detection method. The selected approach takes a rolling average of each phone duration as a percentile of the corresponding range extracted from our research data. If the value of this error is outside the range of normalcy, it indicates an error. Using a rolling average with a length of two phones was found to detect the greatest percentage of errors. Increasing the length of the rolling average was found to decrease the percentage of errors detected but it also increases the number of false positive detections. The cutoff values and the length of the rolling average may be adjusted as needed for different applications. Determination of these variables depends on the relative importance of the percent of errors detected versus the reduction of false positive results.

In addition to the temporal data found within the forced alignments, the signal data procured during the forced alignment is used to detect alignment errors. Our module detects irregularities in confidence scores as an indication of an alignment error. These confidence scores are derived from multiple features in HTK using the 39 PLP coefficients as described earlier. These scores are given for each phone as the average confidence score of the frames within that phone, so these scores must be normalized over the number of frames that make up the phone. The methods used for the confidence score analysis are similar to that of the duration analysis.

For individual phones, a simple rule is run where low confidence scores are flagged. Then, to find problem areas, a rolling average is found and compared against a cutoff. Confidence scores are also determined over entire words by averaging out the values in each individual phone. These combined measures give a robust way for the module to detect errors using the confidence scores.

The next logical step is to combine the temporal and confidence score analyses and advance our heuristic to account for both. As described above, each of these has their own strengths and weaknesses depending on which type of error is present in the transcript. The final module combines the results of both the analyses and uses a support vector machine (SVM) to determine the optimal parameters.

The following image demonstrates the results of the combined heuristic on the example sentence "I have a dream" along with the incorrect transcript "I have a south dream". Each line visually shows the different analyses of the module working. The final result is the word "south" is marked as an error.

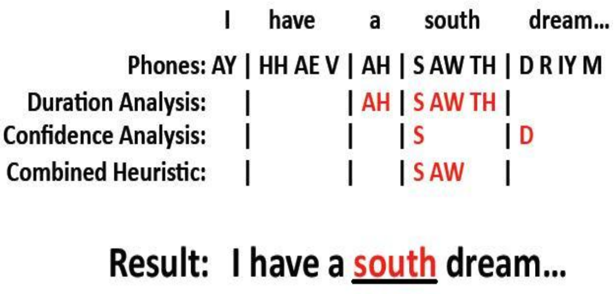
I have a south dream...
Phones: AY | HH AE V | AH | S AW TH | D R IY M
Duration Analysis: | | AH | S AW TH |
Confidence Analysis: | | | S | D
Combined Heuristic: | | | S AW |
Result: I have a south dream...

Figure 7: Example

A support vector machine was used to do a regression analysis of the temporal anomaly and confidence score detection methods. The SVM was implemented using the pyML (python machine learning) package. Regression analysis was used to determine appropriate values for the various cutoff values in the module. These values were then adjusted in order to reduce the false positive results.

5. RESULTS

An experiment was performed in order to determine the efficacy of the error detection module. In order to produce experimental data, one error transcript for each of the three types of transcript errors was generated from each transcript in the TIMIT corpus. These error transcripts were then aligned with their corresponding audio samples. The error detection module was then run on the experimental alignments to provide statistics on the success rate of the system. As there was no other benchmark to compare against, the accuracy of the module had to be compared against the number of false positives that were detected. This would give a better indication of the benefits of the error detection methods. Three different statistics were extracted from this experiment. The first was the percentage of correct hits, which was the number of errors caught out of the total number of errors. The next was the percentage of vicinity hits. Vicinity hits are those hits that mark a word that is not the actual error but directly adjacent to an error word. This was considered because often the aligner would not mark the error right on the word but in between words, particularly in the phone level analysis. A vicinity hit shows the aligner is generally correct, but may have been thrown off by the nature of the error. The final number examined was the percentage of false positives which was the number of correctly aligned sentences marked wrong over all sentence in the trial data set.

The first run was performed using parameters specified by the SVM. The result can be seen in the table below. The total percentage of hits, 81.9%, demonstrated that the module worked fairly well in catching errors. However, the false positives, 5.3%, was alarmingly high for practical applications. If the module incorrectly identified that many errors, its usage would be much more limited. To limit the percent of false positives, more stringent rules were required. In the second trial run, using these new parameters, the percentage of false positives was brought down to 1%. However, this in turn caused the percentage of total hits to also shrink down to 55%. Depending on the application of this module, the parameters can be tweaked as necessary.

	Trial 1	Trial 2
Correct Hits	65.7%	47.8%
Vicinity Hits	16.2%	7.2%
Total Hits	81.9%	55.0%
False Positives	5.3%	1.0%

6. REMAINING WORK

Though the results are generally positive, the improvement of the forced aligner is an ongoing process. One of the immediate improvements that could be looked at are examining energy levels of each phone in the audio file. By looking more into the physics of each phone, a similar analysis can be performed as has been done with duration and confidence score. This can be added to the error detection module to develop an even more advanced heuristic.

Currently, most of the analysis is done on the phone level. Similarly, analysis can be done on the word level. In this case, more attention can be given to semantics. This can be useful for transcripts that were originally hand written.

The implemented temporal error detection methods compares phone durations in an alignment to the durations that we had initially gathered from the TIMIT database. An alternative method would be to perform a hierarchal comparison of the durations of different phones within each word of an alignment. There is ongoing research pertaining to the comparative durations of phones within speech. This data may be used to improve upon the current error detection module.

Additionally, as described in the related work, most common aligners today are fairly accurate when given a near-perfect audio and transcript but not as correct in other situations. This module focuses mostly on transcription errors, but does not specifically examine imperfect audio. Extending the capacity of the module to detect errors in alignment due to errors in the speech signal is necessary in order for forced alignment error detection to be used for most practical applications. Background noise is one of the biggest problem areas for commercial uses of forced aligners. Thus, a poor audio file may need to be separated based on speech and non-speech areas. Furthermore, the forced aligner has to use more complex models to align audio with background noise. This problem lies more in the underlying model of HTK itself. Clearly, each step of the forced alignment process has potential for improvements toward the eventual goal of a robust system that works for a broad set of cases.

7. CONCLUSION

A module for detecting errors in forced alignment has been designed and implemented. It analyzes phone durations and signal data for indications of errors. The results produced in the experiment demonstrate that these methods are effective in detecting errors in forced alignment. While the error detection module produces too many false positive results for commercial applications, the module lays a foundation for further research.

The state of multimedia today dictates the need for forced alignment to increase functionality and accessibility. Unfortunately, while current techniques are passable in certain situation, in adverse conditions the accuracy of forced aligners can be significantly weakened. The error detection module is an important addition to the forced alignment process so that there can be a better understanding of the flaws in current forced alignment techniques. The success of the temporal and confidence score analysis demonstrate that there is potential for a forced aligner to be trained and eventually produce better results.

8. REFERENCES

[1] Timothy J. Hazen. Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Proceedings of the ICSLP*, pages 1606–1609, Pittsburgh, Pennsylvania, 2006.

[2] H. Juang and L. R. Rabiner. Automatic speech recognition–a brief history of the technology development. In *Elsevier Encyclopedia of Language and Linguistics*. Elsevier, second edition, 2005.

[3] Urbanowicz K. and Kantz H. Improvement of speech recognition by nonlinear noise reduction. *Chaos*, 17(2):023121–1–023121–6, 2007.

[4] Lie Lu, Hong-Jiang Zhang, and Stan Z. Li. Content-based audio classification and segmentation by using support vector machines. *Multimedia Systems*, 8(6):482–492, 2003.

[5] Pedro J. Moreno, Chris Joerg, Jean-Manuel Van Thong, and Oren Glickman. A recursive algorithm for the forced alignment of very long audio segments. In *Proceedings of the 5th International Conference on Spoken Language Processing*, Sydney, Australia, 1998.

[6] A.P Varga and R.K. Moore. Hidden markov model decomposition of speech and noise. In *Proceedings of the ICASSP*, Albuquerque, New Mexico, 1990.

[7] J. Yuan and M. Liberman. Speaker identification on the scotus corpus. In *Proceedings of Acoustics '08*.