Proposal Name

Forced alignment of transcripts and audio in a multi-speaker environment.

Aim

The current transcripts corresponding to the videos are both imperfect (with some words missing and some words wrongly transcribed) and lag the audio stream of speech by a variable number of seconds.

The aim of the project is two fold:

a. To obtain **accurate transcripts** in adverse environmental conditions.
b. To **perfectly time align** the obtained transcripts to the videos.

Motivation

The problem of forced alignment is that of matching phonetic segments in an audio sample to their corresponding transcription, which is a vital part of indexing audio/video 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.

Previous Approaches:

A number of methods exist for automatically producing such an alignment from a transcript of an audio source for different types of audio conditions. (**I've given each method in detail at the end in APPENDIX A**)

a. If we have the audio file 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.

b. 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.

c. 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 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.

Alignment in Noisy Environment:

a. One of the earlier methods to tackle noisy environment was to use Hidden Markov Models in signal decomposition as proposed by Varga and Moore. This technique begins with the signal already fully composed and therefore ignores any pre-processing solutions that might be available at an earlier stage.
b. 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.

c. Another technique studied by Urbanowicz and Kantz involved using nonlinear methods to reduce noise by examining the signal beyond second-order statistics.

d. One particular technique proposed classifying audio in five distinct classes: silence, music, background noise, pure speech, and non pure speech. Then, by determining the boundaries between the different sections, it would be possible to handle each case by itself.

e. Lu et al. discovered that using multiple support vector machines, they were able to segment the audio to a high degree of accuracy. While this may be beneficial as a stepping point for certain other techniques, it does not actually clear up the non pure speech.

Shortcomings of the current approaches:

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.

End to end Tools available:

SailAlign
Prosody Lab

Helpful tools available (though not end to end. I've provided the comparison of the techniques in Appendix B):

Praat
CMUSphinx
HTK
Kaldi

Main features of the Implementation:

- Create an alignment tool which does the following:

1) Correct the transcripts and make them more accurate. This includes

a. Compensating for speaker variability.
b. Normalizing background Noise.
c. Dealing with Out of Vocabulary words.
d. Dealing with misspelt and deleted words.

2) Accurately time align the obtained transcripts with the video segment.
3) Documentation of the Project
4) Make it very simple to add a new language in the future.

A Tentative Time line for the Project

I already have an implementation based on **Kaldi (** steps in the implementation covered below in the subsection titled " Steps with Kaldi" **)** ready with the **acoustic models** trained on large corpus which includes all the noises. This should be a good model, and I am open to modifying it as well. Here is the timeline I wish to follow in case selected:

Pre Community Bonding Period

I plan to do the following things till the community bonding period:

April 1 - April 27:
2) There are two ideas which might increase speed as well as accuracy simultaneously using novel signal processing. I'll code them and try to see the improvements if any.

a. **March 30 - April 15:** Duration based approach

b. **April 16 - April 27:** Energy based Approach

Reference: "Additional use of phoneme duration hypotheses in automatic speech segmentation" by Karlheinz Stöber.
I've put a table showing the timeline and the corresponding tasks in brief and then a detailed explanation of the tasks after the table.

BRIEF DESCRIPTION OF TASKS ALONG WITH THE TIMELINE

Time	Task
April 1 - April 15	Try Duration based approach to increase speed. Keep optimizing the already available implementation
April 16 - April 26	Try Energy based approach to increase speed. Keep optimizing the already available implementation
April 27 - May 4	Finetune the models for Spanish and keep running the English alignment with the best method agreed upon after the discussion with the mentors..
May 5 - May 12	Finetune the models for German and keep running the English alignment.
May 13 - May 20	Finetune the models for French and keep running the English alignment.
May 21 - May 27	Finetune the models for Danish and keep running the English alignment.
May 28 - June 4	Finetune the models for Swedish and keep running the English alignment.
June 5 - June 12	Finetune the models for Norwegian and keep running the English alignment.
June 13 - June 20	Correct the errors in English Alignment and keep running the English alignment.
June 21 - June 26	Start the documentation for Speech Recognition Module and tabulate the progress for Mid Term Evaluation.
June 27 - July 4	Start alignment for Spanish and keep running the procedure for English.
July 5 - July 12	Start alignment for German and keep running the procedure on the languages for which the alignment has already begun.
July 13 - July 20	Start alignment for French and keep running the procedure on the languages for which the alignment has already begun.
July 21 - July 28	Start alignment for Norwegian and keep running the procedure on the languages for which the alignment has already begun.
July 29 - August 5	Start alignment for Swedish and keep running the procedure on the languages for which the alignment has already begun.
August 6 - August 13	Start alignment for Danish and keep running the procedure on the languages for which the alignment has already begun.
August 14 - August 21	Finish the documentation.
August 21 - August 28	Bug fixes and finish the documentation.

During Community Bonding Period

- Create a simple interface to define the orthography for all the languages mentioned in the task.

I specifically plan to follow the method used by Simple4All Consortium to obtain the language specific knowledge so that the background knowledge helps in the decoding and alignment.

List of steps with Kaldi:

1) Obtain Language models specific to the language and running the recipe should be fine.
2) The errors , if they come, are due to the lexicon which needs adjusting the text to rectify the same.

Brief Sequence of Modules as per Kaldi:
Data & Lexicon & Language Preparation
Feature Extraction
MonoPhone Training & Decoding ---------- We can stop at this step for speed
Deltas + Delta-Deltas Training & Decoding
LDA + MLLT Training & Decoding
LDA + MLLT + SAT Training & Decoding .... Speaker Adaptive Training
SGMM2 Training & Decoding
MMI + SGMM2 Training & Decoding
DNN Hybrid Training & Decoding
(DNN+SGMM)
DNN Hybrid Training & Decoding
Populating Results File

List of Steps without Kaldi:

This is the standard procedure usually followed.

1) Extract 13 MFCCs along with their first and second time derivatives, giving a feature vector of 39 dimensions.
2) Do Cepstral Mean Normalization for feature normalization.
3) Build cross-word triphone models.
4) Use Phonetic Decision tree tying to cluster triphones. (Derive a set of linguistically motivated questions from the phonetic features defined in the UPS set. The number of tied states, namely senones, can be specified at the decision tree building stage to control the size of the model. The top-down tree building procedure is repeated until the increase in the log-likelihood falls below a preset threshold.)
5) The number of mixtures per senone is increased to 4 along with several Expectation - Maximization iterations. This leads to an initialized crossword triphone model.
6) Relabel the transcriptions using the initialized cross-word triphone models, which were used to run the training procedure once again – to reduce number of mixture components to 1, untie states, re-cluster states and increase number of mixture components.
7) Model the final cross-word triphone with 12 Gaussian components per senone.

I plan to pick the best method from above after discussion with the mentors and use it.

April 27 - May 4 : Finetune the models for Spanish using the best method agreed after the discussion with the mentors.
May 5 - May 12: Finetune the models for German using the best method agreed after the discussion with the mentors.
May 13 - May 19: Finetune the models for French using the best method agreed after the discussion with the mentors.
May 20 - May 27 - Finetune the models for Danish using the best method agreed after the discussion with the mentors.

Post Community Bonding period

Phase 1 (4 weeks):

May 28 - June 4:

- Finetune the models for Swedish using the best method agreed after the discussion with the mentors.
- Simultaneously run the alignment procedure on the English data available. The list of steps below:

1) Obtain audio from the video files
2) Use the acoustic models built to correct the transcripts.
3) Use confidence measures to prune the errors.
4) Align the transcripts with the audio.

June 5 - June 12:

- Correct the errors in the models of different languages
- Simultaneously run the alignment procedure on the English data available.

June 13 - June 20:

- Run the alignment procedure on the English data available using the clusters and correct the minor errors in terms of speed and accuracy.
- Simultaneously start the documentation for the **Speech Recognition** module

June 21 - June 26:

- Finish the alignment procedure for 50% of English data and document the percentages in terms of accuracy.
- Finish the documentation for the **Speech Recognition** module

********** MID TERM EVALUATION **********

Deliverable for Mid Term Evaluation:

- Models ready for all the Languages
- Alignment procedure finished for 50 % of English data.
- A Rough draft of the documentation for the Recognition Module for English.

********** MID TERM EVALUATION **********

Phase 2 (6 weeks) :

June 27 - July 4

- Run the alignment procedure on the Spanish data available using the clusters and correct the minor errors in English data run so far terms of speed and accuracy.
- Simultaneously keep doing the documentation for the **Speech Recognition** module in case left out.

July 5 - July 12

- Run the alignment procedure on the German data available using the clusters and correct the minor errors in data run so far terms of accuracy.
- Simultaneously keep doing the documentation for the **Speech Recognition**.

July 13 - July 20

- Run the alignment procedure on the French data available using the clusters and correct the minor errors in data run so far terms of accuracy.
- Simultaneously finish the documentation for the **Speech Recognition**.

July 21 - July 28

- Run the alignment procedure on the Norwegian data available using the clusters and correct the minor errors in data run so far terms of accuracy.
- Simultaneously start the documentation for the **Language adaptation** module**.**

July 29 - August 4

- Run the alignment procedure on the Swedish data available using the clusters and correct the minor errors in data run so far terms of accuracy.
- Simultaneously start the documentation for the **Language adaptation** module**.**

August 6 - August 13

- Run the alignment procedure on the Danish data available using the clusters and correct the minor errors in data run so far terms of accuracy.
- Simultaneously keep doing the documentation for the **Language adaptation** module**.**

Phase 3 : (3 days)

August 14 - August 17

- Finish the documentation.

APPENDIX A (PREVIOUS WORKS) :

A number of methods exist for automatically producing such an alignment from a transcript of an audio source for different types of audio conditions.

d. If we have the audio file 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.

e. 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]. 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%.

f. 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. 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.

Alignment in Noisy Environment:

c. One of the earlier methods to tackle noisy environment was to use Hidden Markov Models in signal decomposition as proposed by Varga and Moore. 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.

d. 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

c. Another technique studied by Urbanowicz and Kantz involved using nonlinear methods to reduce noise by examining the signal beyond second-order statistics. 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.

APPENDIX B (COMPARISION OF TOOLS) :

As far as the ASR is concerned, here are the three publicly available tools

(i) Sphinx - by Carnegie Mellon University
(ii) HMM Based Toolkit
(iii) Kaldi Toolkit

The Tool Sail Align basically is a wrapper around HTK. It calls the tools of HTK for various steps. Also, I haven't seen papers on Sail Align adding improvement modules to the same.

I've listed the complexity and the speed of the three toolkits available in the table below:

Measure	Complexity (Modification of Scripts)	Speed
Sphinx	Difficult	Slow
HTK	Medium	Medium
Kaldi	Easy	Fast

Coming to the main concern in the project , i.e speed:

Reasons for Less Speed:

HTK runs the recognition using Hidden Markov Models, specifically Context Dependent Hidden Markov Models.
In order to improve the accuracy, it uses iterative algorithm. This means that the algorithm is initially run on to obtain the transcripts, and then these transcripts are used to better the acoustic models. This is called first pass, second pass, etc. This is the reason why in the output, there are 5 files (.lab files) obtained from 5 iterations.

There are three reasons for the speed deficiency in Sail Align (which calls HTK) :

1. It uses Hidden Markov Models
2. It uses iterative models

3. No parallel training.

Steps to increase the speed:

We can choose to increase the speed of the HTK based Sail Align using some tweaks which I'll describe below, or we can choose to build an aligner using Kaldi which is inherently faster due to the application of better models compared to HMMs.
In case we choose to stick to Sail Align, the improvements can come from:

a. **Reducing the number of Iterations:**

Right now the number of iterations are 5. These can be reduced to 3 to improve the speed although there isn't significant improvement when I tried it using 3 iterations.

b. **Constrained Decoding:**

As the task at hand requires the alignment of the first word and the last word to be perfect, specifically and the alignment of the remaining words need not be so accurate (alignment need not be accurate, but the word itself needs to be) , we can make an intelligent move by assigning 5 iterations or more to decode the first and the last word, and then use one iteration to decode the middle words.

c. **Skipping Frames:**

This is in principle similar to the previous approach, but at a much deeper level, by skipping the frames of the middle words as the alignment of the middle words might not be of importance and can be tolerated for speed.

d. **Remove Adaption for already trained segments**

There are other techniques like Beam Search, Limited Decoding, etc. Given that we have a domain, we can achieve in-domain speed optimization using various tweaks as well. However, I personally haven't experimented on them yet.

Steps to improve Accuracy:

a. **Improving the Language Model:**

Language Model acts as the backend for the recognition toolkits and specifies the probability of a word being the one hypothesized. Usually, language models are based on n-grams, meaning that they capture the co occurrence statistics of words and the probabilities depend on these statistics. However, there may be the semantic errors which might arise.

Ex: Will Scholarship and Full Scholarship (from the transcript in the previous mail)

As per co occurrence, both of the above make sense. But as per the meaning of the sentence, only full scholarship makes sense. So, one way to achieve this is to use better language models which don't just rely on the words occurring together, but also their semantic meaning. Neural Network language models are good at this (RNN based)

b. **Improving the acoustic model using Confidence Measure and Lattice Re scoring:**

Simply put, what this means is that if we are using feature A, for decoding, we can use feature B as a 'check measure' and decide a threshold based on B and use it to measure how confident the decoding is.

%%%%%%%%%%%%%%%%%%%%%%

Steps to increase Speed and Accuracy:

Considering that Sail Align is a wrapper on HTK, we can choose to write our own aligner too using Kaldi toolkit which is implicitly faster.

Kaldi uses Deep Neural Networks and LSTMs(Long Short Term Memories) to enhance the speed as well as the recognition accuracy.
I've used Kaldi and its pretty much simple to tweak and make changes and write our own task specific aligner tool using Kaldi.

If indeed we are doing this, we require the following
(If Kaldi, we'll anyway apply the speed and accuracy improvements discussed above) :

a. Voice Activity Detection Module
b. Speech Enhancement Module before Recognition module
c. Recognition Module (includes speaker specific + speaker turns + noise removal, etc)
d. Aligner Module

I've previously modeled all the transient noises for one of my projects(finger snap based authentication). For the same project, I've had to use the Voice activity detection module as well.

APPENDIX C (METHODS USING SIGNAL PROCESSING) :