AIRUS Manual

Pulkit Agrawal, Abhilash Jindal, Nitish Srivastava, Shantanu Agarwal

January 4, 2010

AIRUS: Automated Information Retrieval System Using Speech

1 Introduction

AIRUS produces minutes of meetings, summarizes conferences and discussions using only acoustic information of speech. (i.e. without using any speech to text conversion, thus it is language and vocabulary independent.)

If each speaker is provided with a microphone during the discussion and his voice is recorded, at the end AIRUS would provide a sound file and a corresponding ṗdf file summarizing the important points discussed. AIRUS comes with a user friendly GUI interface including vocal commentary.It is customizable and can adapt itself to the user's notion of important points in a summary.

2 Installation Instructions & Software Requirements

Following softwares are required:

- Matlab R2009a [1]
- Linux OS [2]
- Festival v1.96 or above
- Sphinx [3]
- GCC Compiler
- GNU Make

Following open-source codes have been used:

[1] It has been tested on R2008a and R2009a, probably it would work on other Matlab versions as well

[2] Currently we use Ubuntu

[3] To be integrated in the GUI in future to produce pdf file

- Implementation of YIN Algorithm (Go to Link)
- Implementation of HMM Structured SVM (Go to Link)
- VoiceBox Matlab Toolbox for MFCC calculations (Go to Link)[4] [5]
- C code for R/W Wave files(Go to Link)

For installing, go to the directory containing the folder AIRUS. Run following commands on the command line:
$ cd Airus
$ bash runme.sh

Next follow the instructions given in *Install* of YIN implementation, to get yin.m running.

3 What can AIRUS do ?

- Build models for identifying relevant sections of conversations.
- Use default or a custom trained model for generating summary.

The GUI is self explanatory to carry out these operations.

4 Overview of Working of AIRUS

An overview of step by step functioning of the software is provided here.

4.1 Cross Talk Removal

Cross Talk: The voice of secondary speakers in the microphone of a given speaker.
Voice recording in each microphone is considered, one at a time. Signals with amplitude less than 0.1[6] are set to zero. This completes crosstalk removal.

4.2 Segmenting Sentences

Next, the time when a speaker starts speaking and the time at which he completes his sentence are marked. A speaker is said to have ended his sentence when either of the following 2 occur:

- The Speaker does not speaks anything for one second.

[4] Only downolading melcepst.m & melbankm.m is sufficient
[5] A self written code for the same is also available
[6] assuming that maximum signal amplitude possible is 1.

- While the speaker who was under consideration is silent, some other person starts speaking. Thus we are able to obtain the start and the end times of each sentence[7] by a speaker.

4.3 Definition of Relevancy

A model of relevancy is initially built assuming that we have an annotated data set where the user has manually specified relevant and non relevant segments of the speech. Thus, according to the perception of relevancy of a given user he can build his own dataset over a period of time and can then use AIRUS to build his own model defining what is relevant and what is not. This model can then be used to extract relevant sections of speech from previously unheard conversations. In this sense AIRUS is fully customizable as far as user's perception of what is relevant is concerned.

4.4 Extracting Features

A model of relevancy in our stated problem has to depend only on the acoustic characteristics of speech of like amplitude, energy, distribution of energy in different frequency bands(MFCC) and pitch. Additionally duration for which a speaker is speaking and the order in which speakers are speaking in a discussion also provides cues for determining when someone is going to speak something is relevant.

Currently following features are being used:

4.4.1 40 Mel Frequency Cepstral Coefficients (MFCC)

The average of each of these coefficients is taken over the duration of the sentence. (as MFCC's are calculated for each 30ms window, an averaging operation is done)

4.4.2 Pitch

A sentence is divided into windows of 1 second.(windows are non overlapping)
Pitch of each windowed segment of speech is determined using YIN algorithm.
Thus we get a distribution of Pitch over a sentence.
Mean, Median, Standard Deviation, Kurtosis, Skewness[8] of this distribution are calculated.

[7] Note, the definition of sentence is different from that used in english literature.

[8] Whenever variance = 0, kurtosis and skewness become undefined, thus currently they are actually not a part of feature vectors, 3rd and 4rth order moments can be alternatively considered.

Additionally Pitch over the entire sentence is also seperatley calculated. Thus we get a 4-D vector from Pitch.[9]

4.4.3 Energy

Similar to pitch we get a 4-D vector. Energy is taken as sum of squares of amplitude of signal at each point in a one second window.

4.4.4 Duration of Sentence

4.5 Working with Feature Vector

We have a $40 + 4 + 4 + 1 = 49$ dimensional F.V.
Now Principal Component Analysis (PCA) is used to get a 3-4 dimensional vector. Initial work seems to indicate that performance detoriates if we increase the dimensionality of our F.V.[10]
Also, much better results are obtained if we consider duration seperately. That is we take 48-D F.V., apply PCA to get a 3-D vector and add duration seperately as the 4^{rth} dimension.
The Id of the speaker (like speaker A's Id is 1, Speaker B's id is 2 and so on) forms the 5th Dimension.
We are currently using a 5-D Feature Vector.

4.6 Training and Classification

SVM algorithm is essentially used to train a HMM model using the above features. This trained model is then used for classification.

4.7 Generation of Summary

All relevant sentences (in the correct sequence) are combined into a single audio file to generate a summary. Sphinx can now be used for speech to text conversion.

5 Future Work

5.1 Possiblities in the work done so far

5.1.1 Cross Talk Removal

- Use of ICA algorithm[11] if speech signals from all channels can be exactly aligned.

[9] Ignoring Skewness and Kurtosis.
[10] This can also be a consequence of a very small dataset which we currently had.
[11] Google Blind Source Seperation for more info.

- Making the threshold adaptive and relative instead of an absolute fixed threshold being currently used.

5.1.2 Windowing in Pitch and Energy features

The window size can be varied and overlapping windows can be used. This might further improve the results. Ideally, a cost function should be written to find the optimal size and overlap of windows.

5.1.3 MFCC Features

Instead of averaging, we can form a distribution of MFCC similar to calculation in Pitch and Energy. Also the difference and diference of difference MFCC's have not been considered yet.

5.1.4 SVM trained HMM Model

Some parameters of SVM for training have been heuristically chosen. Proper evaluation for optimal choice needs to be made.
Also we beleive that this algorithm aims at improving the accuracy of the calssification, some time can be spent on ways to tweak the inherent cost function on which the SVM trains the HMM model so as to improve the recall with acceptable tradeoff in accuracy.

5.2 Other Models for Classification

- Incorporating sequence information in SVM itself, thus doing away with HMM altogether.
- Use of string prediction.[12]

6 Bottom Line

Currently, the features which are most relevant are lenght of a sentence and the order in which speakers are speaking. When we add other acoustic features we get an improvement of $\tilde{1}0\%$ in recall. Untill now, acoustic information without sequence and length of senetence seems to provide no information by itself. One more key issue is how would a model trained on say n speakers in a discussion generalize to the case when m speakers participate.

The preliminary results are very promising. Since, the dataset was very small no firm conclusions can yet be drawn.

[12] Refer to slides for illustraion.

7 Copyright