Technical Report

VESTA: Audio Processing Services

Final version
CRIM-15/12-18/RECO

Gilles Boulianne
Senior Counsellor
Speech Recognition
December 18th, 2015

Financial partner:
## Contents

1 Introduction .............................................. 3

2 Speaker diarisation .................................. 3
   2.1 Technical description .................................. 3
   2.2 Experiments and results ................................. 4
       2.2.1 Noise reduction .................................. 4
       2.2.2 Deep Learning based speaker change point detection .......................... 5
   2.3 Evaluation ............................................. 6

3 Speech to text ......................................... 7
   3.1 Technical description .................................. 7
   3.2 Evaluation ............................................. 8

4 Speech and Text matching .............................. 9
   4.1 Technical description .................................. 9
   4.2 Evaluation ............................................. 10
1 Introduction

This report describes the technical aspects of the audio processing services that are part of the software platform known as VESTA (Video Evaluation System for Task Analysis).

2 Speaker diarisation

This service partitions the input audio recording into homogenous segments according to the speaker identity, in effect finding "who spoke when". Over the entire audio recording, segments belonging to the same speaker are clustered together and labelled with a unique identifier for the recording. The service provides information about speech absence or presence, speaker turns, speaker identity, and speaker gender.

Figure 1 illustrates how the original signal (bottom plot) is split into several segments, each having a color identifying a particular speaker. Speaker label appears at the top right of the figure.

![Figure 1. Original signal (bottom) and diarisation result (top).](image)

2.1 Technical description

This algorithm was adapted from our earlier work on wideband audio [1]. It is a multistage algorithm, as illustrated in Figure 2, and combines clusters from Gaussianized and non-Gaussianized MFCC features [2].
2.2 Experiments and results

2.2.1 Noise reduction

During user tests, some input files caused catastrophic failures of the diarisation process. It turned out that the background noise level in these files was so high that the whole file would be classified as speech by the Voice Activity Detector, and this in turn caused the algorithms to fail for very noisy files longer than one hour.

To make diarisation more robust to such noisy input, we explored several techniques, keeping in mind that any chosen solution must not require training on specific data, and must not degrade current performances. Bandpass filtering to keep only frequencies of interest for speech (300 Hz to 3400 Hz) was not enough to solve the problem. Wiener filtering did not help either.
Our final solution involves measuring the noise level inside a sliding window of 25 ms, advancing in 10 ms steps. A power spectrum is computed for each 10ms frame, and we estimate the signal-to-noise ratio using the Wiener entropy of the energy distribution across frequencies. The log value of this estimate is smoothed over a 25 frame window, and normalized to a maximum value of 0 for the entire file. The smoothed log is exponentiated and resampled to provide a weighting scalar between 0 and 1 that is applied to each individual sample in the time domain. This technique does not introduce audible artefacts, and the use of the Wiener entropy makes it robust over a large variety of noises. Figures 3 and 4 show spectrograms of an original recording and its cleaned-up version, respectively. After noise reduction, the recording can be submitted to the same diarisation algorithm. For clean files, there is no effect on diarisation.

![Figure 3. Spectrogram of original, noisy recording.](image)

![Figure 4. Spectrogram of recording after noise reduction.](image)

### 2.2.2 Deep Learning based speaker change point detection

Audio or speaker change point detection is the process of locating time points (or frames) in an audio stream that correspond to a transition from one speaker to another, or from music to speech or vice-versa. It is the second step in diarisation, as illustrated in the top part of Figure 2.
We investigated the use of deep neural nets (DNN) to provide initial speaker change points in a speaker diarization system. We trained DNN states that correspond to the location of the speaker change point (SCP) in the speech segment input to the DNN. A confidence threshold on the DNN output value decides if a change point is detected or not. We showed that this DNN-based change point detector significantly reduces the number of missed change points for the English test set, as shown in Table 1, when compared to the conventional KL2-based detector, as shown in Table 2, when operating at a similar false alarm rate.

<table>
<thead>
<tr>
<th>conf thresh</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>% missed</td>
<td>11</td>
<td>12</td>
<td>16</td>
<td>24</td>
<td>31</td>
<td>43</td>
</tr>
<tr>
<td>false alarms/min</td>
<td>29.3</td>
<td>26.9</td>
<td>22.7</td>
<td>18.5</td>
<td>13.7</td>
<td>9.6</td>
</tr>
</tbody>
</table>

Table 1: Percentage missed and false alarm rate/min for DNN-based changed point detector, English test set.

<table>
<thead>
<tr>
<th>% missed</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>false alarms/min</td>
<td>33.0</td>
</tr>
</tbody>
</table>

Table 2: Percentage missed and false alarm rate/min for KL2-based changed point detector, English test set.

Even given these large improvements in the speaker change point detection, when we measure the overall DER on the English test set, it only goes down from 12.91% (for KL2-based CPD) to 12.51% (for DNN-based CPD). So work remains to be done before this can be integrated into the current diarisation system. These results were published in [3].

### 2.3 Evaluation

The main metric of performance is the diarization error rate (DER) as defined by NIST in the RT-04 Fall evaluation [4]. The DER is the sum of three errors: missed speech (speech in the reference but not in the hypothesis), false alarm speech (speech in the hypothesis but not in the reference), and speaker match error (reference and hypothesized speakers differ). We used the md-eval-v17.pl Perl script from NIST to estimate this DER.

The diarisation error rate on a test corpus of pairs of telephone conversations (so there are 4 speakers per test file) is between 6.7% and 10%. See [1] for more details about the evaluation procedure.
3 Speech to text

This service automatically transcribes speech found in the audio recording into text. It is a state-of-the-art system using Deep Neural Networks. In order to be useful for a wide range of uses, it was designed to work well with any speaker (speaker-independent) and can accommodate recordings with varying bandwidths, from telephone to wideband. As a first step, it applies diarisation (see previous section) to identify speech segments and speaker turns.

3.1 Technical description

The speech-to-text is an HMM-DNN recognition system, with code based on the Kaldi toolkit [5]. The overall design of the system is similar to the CRIM system described in [6], but the training database, acoustic and language models were developed specifically for VESTA and the differences are reported here.

Training data. A total of 1,142 hours of transcribed telephone speech were used to train the system, including half the Fisher corpus [7], Switchboard [8] and downsampling versions of the Hub4 [9], WSJ [10], RT03, RT04 [4] and Marketplace [11] data.

Features. CRIM uses TRAP (TempoRAI Pattern) features [12] extracted from filterbank as input to the neural net. To compute TRAP features, 23-dimensional filterbank features are normalized to zero mean per speaker. Then 31 frames of these 23-dimensional filterbank features (15 frames on each side of current frame) are spliced together to form a 713-dimensional feature vector. This 713-dimensional feature vector is transformed using a Hamming window (to emphasize the center), passed through a discrete cosine transform and the dimensionality is reduced to 368. This 368-dimensional feature vector is globally normalized to have zero mean and unit variance.

Acoustic modelling. For training the deep neural network (DNN) using back propagation, 3 hours of the training audio were set aside for validation. The DNN has 7 layers, with an input layer of size 368, 5 hidden layers of size 1,000 and one softmax output layer with 3,925 outputs. The alignments and output labels for training the DNN come from the GMM-HMMs with 3,925 states and 400k Gaussian means (trained with MLE only). Training was done by back propagation with a cross-entropy objective function, followed by two iterations of alignment of training data with the resulting DNN-HMM and retraining of the DNN using the aligned data, then 2 iterations of sequence training using an MMI criterion.

Language modelling. A trigram language model trained on 3 million words from Switchboard and CallHome training transcriptions, with 1.4 million bigrams and 1.6 million trigrams. The vocabulary of 30,000 words was selected from the most frequent words in the training set. This language model has a perplexity of 90 on data held-out from the training set.
3.2 Evaluation

The word error rate of the speech to text service was evaluated on the Switchboard eval2000 test set [13], which combines Switchboard and CallHome data. The test set contains 42,989 words, and the overall word error rate on the combined set is 24.3% (17.1% on Switchboard and 27.0% on CallHome).
4 Speech and Text matching

This service main goal is to detect, in the audio recordings, passages that are exact citations of text that can be found in a collection of reference documents. The original use was to monitor "think aloud" utterances of medical students in order to find out if they were citing their manual when looking for diagnostics. However, it is general enough to serve in several speech and text alignment applications. It could be used, for example, to speed up correction of television closed-captioning before a program is aired again, by find intervals where text exactly matches the recorded dialogue, so they don’t need to be corrected and can be skipped altogether. It has also been used for cleaning freely available databases such as Librivox for training speech recognition systems [14].

The service reports the best matches between speech in the recording and text in the reference documents. The matches are returned in ranked order, with longest matches first. Each match is identified by its span in the audio and the corresponding span in the text document, as illustrated in Figure 5.

![Figure 5. Matching spans in the audio and in the reference documents.](image)

4.1 Technical description

The service relies on the same acoustic models and speech recognition engine as the speech to text service. However, in speech to text, any word sequence is possible. For the matching, the possible word sequences to be scored by the recognition engine are described using factor automatas [15].
The idea is to encode all possible sequences of consecutive words that are found in the reference documents. These include all sequences of $N$ words, for $N = 1$ to $N = L$ where $L$ is the document length; these sequences are called "factors" of the whole document. A factor automaton yields a compact representation of these sequences, and result in an efficient alignment algorithm. The basic structure of the factor automaton is shown in Figure 6.

![Factor automaton for a text $w_1, w_2, ..., w_n$.](image)

Because the audio segment that matches a word sequence can occur anywhere in the middle of a long audio recording, the factor automaton must allow the matching word sequence to be preceded and followed by arbitrary word sequences. This is obtained with epsilon transitions (labelled with #0:#0 in Figure 6). The automaton also allows several matches to be present in the same audio recording, thanks to the transitions labelled <sil>:<sil>. In practice, transition labels are augmented with additional information to identify the document and word starting and ending positions in the document. Penalty weights are also added for insertion and deletion transitions so that longest matching sequences are encouraged.

Such an automaton, build with the whole document collection, can then be used as a speech recognition grammar. Thus the speech to text engine will return, for each audio recording, a word sequence interspersed with #0 symbol. Each subsequence that appear between two #0 symbols is an exact citation from one of the document in the collection and constitutes a potential match. Because the speech to text engine reports beginning and ending times for each output word, each match also corresponds to an audio segment in the recording.

### 4.2 Evaluation

The matching algorithm was evaluated with an audio book from LibriVox, an English recording of "Count of Monte Cristo", from Alexandre Dumas. We used 18 chapters (from 100 to 117). Each chapter was read by a different speaker. Diarisation was used to split the long audio files in 1,476 short audio segments of various lengths, from 1 to 300 seconds, with an average duration of 15.5 seconds. This corresponds to the intended use of the service, where diarisation would first be performed on a long audio recording, and individual segments used for look up in the
reference text collection.

The evaluation is set up as a document retrieval task. Each short audio segment is used to query the chapter collection. The algorithm returns a set of matches in the form of text spans, i.e. word sequences with chapter and beginning and ending position in that chapter. In order to measure false alarm (FAR) and false rejection rates (FRR), we designate one of the chapter (103) as the target. This yields a test set with 45 target queries (audio segments from chapter 103) and 1,431 impostors (audio segments from other chapters). A match is deemed correct if both the audio segment and the text span are from chapter 103. It is a false alarm if the audio segment is not from chapter 103 but the text span is. It is a false rejection if the audio span is from chapter 103 but the text span is not.

Each audio segment is assigned a "matching" score based on the number of words in the matches and the duration of the matches relative to the audio segment duration. The decision is to accept an audio segment as a citation when the score exceeds a threshold. Figure 7 shows the DET curve we obtain when varying this threshold from the minimum value (producing highest false alarm rate) to the maximum value (producing highest false rejection rate).

![DET curve](image)

**Figure 7. DET curve for 17 chapters of "Count of Monte Cristo".**

The EER (Equal Error Rate), which the point on the DET curve where FAR=FRR, is 8.22%.
References


