ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

AN IMPROVED PREPROCESSOR FOR THE AUTOMATIC TRANSCRIPTION OF BROADCAST NEWS AUDIO STREAM

Jindrich Zdansky, Petr David, Jan Nouza

Department of Electronics & Signal Processing
Technical University of Liberec, Hálkova 6, 461 17 Liberec 1, Czech Republic
{jindrich.zdansky, petr.david, jan.nouza}@vslib.cz

10.21437/Interspeech.2004-373

Abstract

This paper deals with the preprocessing of the broadcast news (BN) audio stream for the automatic transcription purposes. The preprocessing consists of the automatic segmentation followed by the broad-class segment identification. The former is capable of detecting speaker and/or acoustic changes in the BN audio stream with the precision being 82.75%. The latter acts as a filter that removes non-speech parts. The performance of the proposed system was evaluated on the multi-lingual pan-European COST278 BN database containing data in 6 languages. The preprocessing and segmentation module operates in a near-real-time way, with the total delay of 12 seconds. Its practical functionality was evaluated on the Czech part of the BN database. The automatically segmented signal was directly sent to the large vocabulary speech recognition system operating with a 200K-word Czech lexicon. The difference in performance between automatically and manually segmented BN streams was only minimal - 1.12%.

1. Introduction

Broadcast news transcription is one of the most challenging tasks in the area of automatic speech-to-text processing. Since the task is extremely difficult and its solution demands complex software tools (namely the decoding engine), most systems reported in literature are based on well established platforms developed at several leading labs like, AT&T, BBN, LIMSI or IBM. This is not the case of our system that has been completely built in our lab during the last 3 years and now it is capable of transcribing spoken Czech using a 200k lexicon with the WER being in the range of 15 - 30% depending on the speaking style and environment noise. The recognition system is described in another paper submitted to this conference.

Here we want to focus on its preprocessing module. Its task is to take the complete audio stream and segment it into acoustically homogenous parts belonging to individual speakers or environment conditions. Besides the main function (BN transcription) the output of this system could be used for other purposes e.g. for searching through audio and video clips, indexing video conferences, etc.

2. System overview

The data flow diagram of the proposed unsupervised BN transcription system is illustrated in Fig. 1.

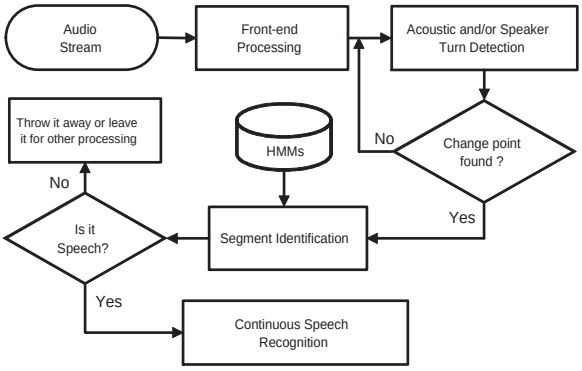


Figure 1: *Diagram of automatic transcription system.*

A compact 16 kHz audio stream from arbitrary source is the input to the front-end processing module, which converts the input audio into feature vectors (first 13 MFCCs, 10/25ms) that are continuously delivered to the component which provides acoustic and/or speaker change detection (see sect. 4). On the generated, acoustically homogenous segments, the cepstral mean subtraction is applied. These blocks of data are sent into the component providing segment identification (see sect. 5) to perform speech/non-speech detection. Finally, the acoustic feature vectors are supplemented by their delta and acceleration coefficients and these data blocks are passed to the large vocabulary continuous speech recognition system (see sect. 6).

3. Multi-lingual TV BN database

The data used in our experiments have been collected as a part of a pan-European Broadcast News Database by 6 institutions collaborating in the European COST278 action on Spoken Language Interaction in Telecommunication. At present, the database comprises 6 parts: Slovak (SK), Slovenian (SI), Galician (GA), Portuguese (PT), Dutch (BE) and Czech (CZ). Each participant has prepared 3 hours of its national complete news broadcasts from public and/or private

TV stations. The data set consists of wav files (16 kHz, 16 bit, mono), video files and transcription files. Each set is divided in two parts: two hours for the development and one hour for the testing purposes. More detailed description could be found in the paper [4].

4. Audio stream segmentation

The segmentation of continuous audio stream is useful as a preprocessing stage for further classification of the segments. It allows for e.g. noise rejection, music removal or HMMs adaptation for more accurate automatic transcription of BN.

4.1. Acoustic Changes Detection

In this paper we present our approach that is based on *Bayesian Information Criterion* (BIC).

4.1.1. Brief review of BIC theory

The key idea of the audio signal segmentation via BIC is to view the change detection task as a model selection problem. BIC is a likelihood information criterion allowing choosing among a set of candidate parametric models M_k to describe a given data set appropriately. This criterion is defined as

$$BIC(M_k) = \log \mathcal{L}(X|M_k) - \frac{\lambda}{2}\,\mathcal{C}(M_k)\log N \qquad (1)$$

where $\mathcal{L}(X|M_k)$ is the maximum likelihood term, $\mathcal{C}(M_k)$ denotes number of free parameters of the model and the λ is the penalty weight. According to BIC theory, the best model is the one with the highest BIC.

Considering the input audio signal to be multivariate Gaussian process in the cepstral space, we can estimate maximum likelihood (ML) parameters of the Gaussian model $x_i \sim N(\mu_1^N, \Sigma_1^N)$ to describe data set $X_1^N = \{x_i \in \Re^d; i = 1, \ldots, N\}$, where μ_1^N denotes sample mean, Σ_1^N sample covariance and d is the dimension of the feature vectors. Assuming Gaussian model M, formula for computing ML term can be determined as

$$\log \mathcal{L}(X_1^N|M) = -\frac{N}{2}\,\log|\Sigma_1^N| - \frac{d}{2}\,N\log(2\pi) - \frac{N}{2} \qquad (2)$$

and the complexity term can be enumerated as

$$\mathcal{C}(M) = d + \frac{1}{2}\,d(d+1). \qquad (3)$$

4.1.2. One change point detection

For the simplicity, consider the one change point i in the data set X_a^b. We can make simple test to make decision if i is really the change point by comparing BIC of two models. First one is the single Gaussian model M_1 estimated on the data X_a^b and the second one is the double Gaussian model M_2. First Gaussian of M_2 is estimated on the data X_a^i and the second one on the data X_{i+1}^b. If BIC difference $\Delta BIC = BIC(M_2) - BIC(M_1)$ is positive, then there is apparent advantage in modeling data by double Gaussian, which signifies the change point. Using eq. (1) and (2), final formula for ΔBIC can be determined

$$\begin{aligned}\Delta BIC(i) = {} & \tfrac{b-a+1}{2}\log|\Sigma_a^b| - \tfrac{i-a+1}{2}\log|\Sigma_a^i| + \\ & - \tfrac{b-i}{2}\log|\Sigma_{i+1}^b| - \tfrac{\lambda}{2}\mathcal{C}(M)\log(b-a+1).\end{aligned} \qquad (4)$$

Location of the ML changing point p can by obtained by solving the equation $p = \arg\max_i BIC(i)$, where $i = a+B, \ldots, b-B$; and B is the least number of feature vectors to make a good estimate of the covariance matrix.

4.1.3. Multiple change points detection

There are two different approaches for multiple change point detection - adaptive window length scenario and fixed window length scenario. The first one presented by the IBM researchers [1] seems to be very elegant, because does not require adjustment of any free parameters. But it suffers from low stability, because success of change point detection is dependent on the correctness of the previous one detection.

The second approach uses overlapping fixed window length scenario, which seems to be more robust, but always needs to adjust some thresholds or penalty weights. In this scenario one gets some uneven curve (whatever method we use) with the peaks localizing the maximum probable changing points in the context of the window. The task is to find these peaks and decide if it is really true change point. Several solutions have been suggested to create this curve, like *Normalized Log-Likelihood Ratio* [3] or various combinations of several kinds of distance measures [2]. In the advanced segmentation systems this so-called *generation stage* is supplemented by the *rejection stage*, often realized as another variation on the Bayesian theme.

4.2. Proposed Segmentation Algorithm

In this section we describe our fixed-length window technique capable of fast and accurate segmentation, which uses ΔBIC for both the generation and rejection stage. In the generation stage, a fixed length window ΔBIC is computed for every frame of the input feature stream. Then the peaks of the ΔBIC curve are detected and are considered as boundary candidates. The generation stage (controlled by 3 free parameters) is supplemented by the rejection stage (controlled by another free parameter). Free parameters are denoted as α, β, L_G, L_M.

4.2.1. ΔBIC computation

For each input feature vector $i > L_G$, the value of $\Delta BIC(i)$ is computed according to eq. (4), where $a = i - L_G - 1$, $b = i + L_G$ and the penalty weight $\lambda = \alpha$. The variable L_G can be viewed as a delay, which is needed to be implemented, so that the system becomes causal. An example of ΔBIC curve is shown in Fig. 2.

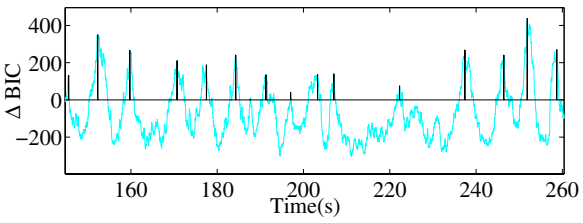


Figure 2: ΔBIC *curve with automatically marked peaks denoting the segment boundary candidates.*

4.2.2. Rough estimate of segment boundaries

To get the segment boundary candidates we used the following approach. Consider the FIFO buffer BF of the length of L_M initialized with zeros. Each $\Delta BIC(i)$ value produced by the previous stage is pushed into the buffer. Then, the maximum in this buffer $MX = \max_k BF(k)$ and its position $mx = \arg\max_k BF(k), k = 1, \ldots, L_M$ are found, and the equation

$$BF(mx) = BF(mx) + MX \tag{5}$$

is calculated. After that, the condition

$$\frac{1}{L_G}\left(BF(L_M) + |BF(L_M)|\right) \geq |\Delta BIC(i+L_M-1)| \tag{6}$$

is checked and if fulfilled, frame $i + L_M - 1$ is considered to be a proper boundary candidate $c(n)$. An example of the peaks revealed by this method is shown in Figure 2. This system can become causal by implementing the delay L_M.

4.2.3. Rejection Stage

The generation stage produces boundary candidates $c(n)$ with the delay $L = L_G + L_M$. This delay and the previously accepted boundary position t can be used for the confirmation or rejection of the candidate boundary $c(n)$. At this moment we know the rough estimate of the position. Eq. (4) can be used for increasing accuracy of detection and also for confirming or rejecting the boundary candidate $c(n)$. Using the data $X_t^{c(n)+L_G+L_M}$ and by solving equation

$$c(n) = \arg\max_k \Delta BIC(k) \tag{7}$$

$k = c(n) - A, \ldots, c(n) + A$, we get more accurate boundary candidate position. If $BIC(c(n)) > 0$, then the candidate is accepted, failing that, it is rejected. Variable A is the neighbourhood which should be examined, $A \sim 0.5-1$ second is reasonable setting. The rejection stage can be controlled by the penalty weight β.

4.3. Segmentation system performance

The algorithm described in the previous section was evaluated on the testing part of the database (see sect. 3). We count a hit if the interval between the computed and real boundary is shorter than 1 s. Then we determine

$$faultrate = \frac{I+D}{N}, precision = \frac{H}{H+I}, recall = \frac{H}{N} \tag{8}$$

where $H,\ I,\ D$ are the numbers of hits, insertions, deletions, respectively and N is total number of the true boundaries. To estimate the 4 free parameters of the segmentation system we used all 12 hours of the training data and the optimal setting was achieved for penalty weights $\alpha = 1.8, \beta = 2.6$ and for delays $L_G \sim 4s, L_M \sim 8s$.

Table 1 summarizes the results for each the 1-hour length language components included in the database. Because most of the acoustic changes in the database are speaker turns, we have evaluated the speaker segmentation results. The performance of the system can be further increased by tuning the free parameters separately for each language. Then average fault rate will decrease from 36.46% to 32.52%.

Language	Recall [%]	Precision [%]	Faultrate [%]
SK	92.37	76.10	36.64
SI	77.91	78.88	42.94
PT	88.58	89.40	21.92
GA	92.97	75.80	36.72
BE	75.76	81.08	41.92
CZ	67.88	89.91	39.74
ALL	80.28	82.75	36.46

Table 1: *Speaker segmentation results.*

Although the computational load of the proposed segmentation technique is small (incl. parametrization it takes 10% computer time on a PC with 2.6 GHz processor) it could be further reduced by possible frame-dropping. The maximum delay is data dependent and can be determined as $L = L_G + L_M$. In the case of our BN data it was 12 seconds.

5. Segment Identification

The segmentation module is followed by an identification stage that is to avoid passing non-speech segments to the speech recognizer. The speech/non-speech detection is done by a probabilistic classifier that distinguishes between five broader classes: speech (clean, speech + background music, speech + background noise) and non-speech (music, noise sounds). Various classifiers are commonly used: Gaussian Mixture Models, k-Nearest-Neighbors, Neural Networks, Hidden Markov Models (HMM), etc. Our system employs the last one.

In the evaluation we used three types of 64-mixture HMMs: 1-state, 3-state and 5-state ergodic models. They have been trained using the Baum-Welch algorithm on the training part of the BN database. The Viterbi algorithm was used to classify the segment class.

The percentages of correctly labeled speech/non-speech frames are listed in Table 2. The last line in the table (denoted as OFF) compares our results with those that would be

#	1 state		3 state		5 state	
Lng	spch	non	spch	non	spch	non
SK	96.62	45.98	93.60	56.93	98.09	56.68
SI	98.73	78.19	99.68	86.58	99.61	89.31
PT	99.97	84.71	98.46	83.81	99.33	83.81
GA	99.31	45.41	97.95	49.81	99.53	49.07
BE	99.60	32.64	97.66	25.86	99.36	26.80
CZ	99.05	83.34	95.79	88.12	99.05	83.34
ALL	98.86	58.79	97.16	63.42	99.17	63.43
OFF	96.07	89.39	96.36	83.33	98.45	77.89

Table 2: *The percentage of correctly labeled speech/non-speech frames when using 1, 3, 5 state HMMs.*

obtained if the Viterbi decoder is applied off-line, i.e. before the segmentation. It is evident that our scheme, in which speech/non-speech detection follows speaker/acoustic segmentation, has slightly better performance. The application of the best (5-state ergodic) HMM revealed 99.17% speech frames and 36.57% non-speech frames were wrongly classified as a speech. But the latter errors do not have so serious impact. They will just cause that these frames will be send to the speech processing module, which will be still capable of handling the remaining noise events.

6. Automatic BN transcription

This section only briefly describes the speech recognition experiments we performed on the Czech component of the BN database using our *Large Vocabulary Continuous Speech Recognition System* (LVCSR) - for details see [5].

6.1. Speech Recognition System

Our LVCSR system employs an optimized version of the one-pass time-synchronous Viterbi search. Its acoustic part uses 48 HMMs, from these 41 represent the Czech phonemes, the remaining 7 models belong to noise events. All the models are context-independent but they use very large numbers of gaussian mixtures (up to 100). The whole HMM inventory thus include 144 different model states with the total number of 12,280 Gaussians. For the BN task we employ the 200k words multiple-pronunciation vocabulary together with the corresponding bigram language model smoothed by Witten-Bell method. With this lexicon we are able to achieve the Out of Vocabulary (OOV) rate in the range of 1-2% on different newspaper text corpora.

6.2. Evaluation of the complete system chain

The Czech data was passed to the complete BN transcription system depicted in section 2. We evaluated and compared the recognition results that were obtained with manually and automatically segmented news streams. The results are shown in Table 3.

From the above results we can see that the automatic segmentation introduces a small error, which is most probably due to possible inside words/sentence segmentation cuts. Note that the recognition results have been evaluated an all news data, including those parts with background music or loud street noise.

Preprocessing	Accuracy[%]	Correctness[%]
Automatic	69.20	72.74
Manual	70.32	73.66

Table 3: *Results from the Czech BN transcription task: automatic vs. manual segmentation and preprocessing*

7. Conclusions

We have proposed quite fast and accurate preprocessor for the automatic transcription of streamed BN audio signal. Its performance is comparable with the manual segmentation, which is known to be very tedious and time consuming. The automatic segmentation module operates with time delay about 12 s, which is acceptable for many applications and seems to be faster than some other techniques published in literature. The preprocessor works with the 82.75% precision with respect to the speaker turn detection and is able to reveal 99.17% of the speech frames, whereas 36.57% of non-speech frames are wrongly passed to the speech recognition module. These results have been obtained on a multi-lingual database of broadcast news in 6 European languages.

8. Acknowledgement

This research was supported by the grant COST OC 278.02.

9. References

[1] Chen, S., Gopalakrishnan, P. S., “Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion”, Proc. of DARPA Broadcast News Transcription and Understanding Workshop, 1998.

[2] Lie Lu, Hong-Jiang Zhang, “Speaker Change Detection and Tracking in Real-Time News Broadcasting Analysis”, Proc. of the tenth ACM international conference on Multimedia, Juan-les-Pins, France 2002.

[3] Vandecatseye, A., Martens, J. P., “A Fast, Accurate and Stream-Based Speaker Segmentation and Clustering Algorithm”, Proc. of Eurospeech, Geneva, 2003.

[4] Vandecatseye, A. et al, “The COST278 pan-European Broadcast News Database”, Proc. of the fourth International Conference on Language Resources and Evaluation, Lisboa, Portugal, May 2004.

[5] Nouza, J., Nejedlova, D., Zdansky, J., Kolorenc, J., “Very Large Vocabulary Speech Recognition System for Automatic Transcription of Czech Broadcast Programs”, Submitted to ICSLP 2004.