Segmentation of Telephone Speech Based on Speech and Non-speech Models

Michael Heck, Christian Mohr, Sebastian Stüker, Markus Müller, Kevin Kilgour, Jonas Gehring, Quoc Bao Nguyen, Van Huy Nguyen, and Alex Waibel

Institute for Anthropomatics, Karlsruhe Institute of Technology, Germany
{heck,christian.mohr,sebastian.stueker,m.mueller,kevin.kilgour, jonas.gehring,quoc.nguyen,van.nguyen,waibel}@kit.edu

Abstract. In this paper we investigate the automatic segmentation of recorded telephone conversations based on models for speech and non-speech to find sentence-like chunks for use in speech recognition systems. Presented are two different approaches, based on Gaussian Mixture Models (GMMs) and Support Vector Machines (SVMs), respectively. The proposed methods provide segmentations that allow for competitive speech recognition performance in terms of word error rate (WER) compared to manual segmentation.

Keywords: support vector machines, segmentation, speech activity detection.

1 Introduction

Speech recognition in telephone calls is still one of the most challenging speech recognition tasks to-date. Besides the special acoustic conditions that degrade input features for acoustic modelling, the speaking style in telephone conversations is highly spontaneous and informal. Each channel of a conversation contains large parts with no speech activity. Assuming equal participation of both speakers in the conversation, at least 50% per channel can therefore be omitted for recognition. Omitting non-speech segments on one hand improves recognition speed and on the other hand can improve the recognition accuracy since insertions due to falsely classified noises in the non-speech segments can be avoided, which is especially promising in the variable background noise conditions of telephone and mobile phone conversations.

We investigate two methods of automatic segmentation to determine sentence like chunks of speech and filter out non-speech segments for speech recognition. As a baseline we regard the segmentation on the output of a regular speech recognizer. Our experimental setups make use of a GMM based decoder method and an SVM based method.

Evaluation is done according to speech recognition performance since references for speech segments are not very accurate. The evaluation took place on corpora of four distinct languages, that were recently released as the IARPA Babel Program [1] language collections. babel106b-v0.2f and the subset babel106b-v0.2g-sub-train cover *Tagalog* and are used in two training data conditions, *unlimited* and *limited*, respectively. In the *unlimited* scenario, a full data set covering approximately 100 hours of transcribed audio material was available for training, whereas for the *limited* case only a subset of the

M. Železný et al. (Eds.): SPECOM 2013, LNAI 8113, pp. 286–293, 2013.

available data was approved for training, comprising approximately 10 hours each. The additional three languages collections used for the *limited* case are babel101-v0.4c for *Cantonese*, babel104b-v0.4bY for *Pashto* and babel105b-v0.4 for *Turkish*.

The outline of the paper is structured as follows. Sections 2, 3 and 4 describe the segmentation methods we used. In Section 5 the handling of the training data for the speech/non-speech based methods is described. Evaluation is shown in Section 6 and Section 7 concludes and points out future work.

2 Baseline

For the baseline automatic segmentation a fast decoding pass with a regular speech recognition system on the unsegmented input data is done to determine speech and non-speech regions as in [2]. Segmentation is performed by consecutively splitting segments at the longest non-speech region with a minimal duration of at least 0.3 seconds.

Like all HMM based systems addressed in this paper the speech recognition system used for decoding was trained and tested using the JANUS Recognition Toolkit that features the IBIS single pass decoder [3]. The system employs left-to-right HMMs, modelling phoneme sequences with 3 HMM states per phoneme.

3 GMM Based Decoder Method

Since for segmentation the classification problem only consists of two classes, namely speech and non-speech, in the GMM-based method we use the same Viterbi decoder as in the baseline method and use GMM models for speech and non-speech. We found that splitting the non-speech model into a general non-speech model and a silence model increased performance. Our HMM segmentation framework is based on the one in [4] which is used to detect and reject music segments. This approach was also used in [5] for acoustic event classification. Similar approaches for the pre-segmentation of very long audio parts for speech recognition systems were used in [6] where GMM models were trained for speech, speech + background music, non-speech noise, music and pause. Alternatively a phoneme decoder using regular phoneme models and a phoneme bi-gram model is investigated. HMM based segmentation of telephone speech was also presented in [7].

We use MFCCs with 13 coefficients and its delta and double delta as input features. Window size is 16 milliseconds with a window shift of 10 milliseconds. We tested additional features such as a zero crossing rate, but it did not improve performance. We also tried to stack the MFCC plus delta and double delta features of both audio files for each call to take into account that – neglecting parts of cross-talk – if a segment contains speech in one channel, the other channel does not. However, audio files of both the training and test data set were not synchronised channel-wise, so that the dual channel models decreased performance.

A-priori probabilities are modelled as 2-grams but we assume equal probability for all segments and 2-grams since we handle each telephone call as two channels, one for each speaker, and assume both speakers have the same contingent in the conversation so at least half of each file contains non-speech segments.

All types of segments are modelled as single HMM states, with the minimal segment durations being modelled directly by the HMM topology. For speech segments the minimal duration is 250 milliseconds, for non-speech segments 150 milliseconds. Each GMM consists of 128 Gaussians with 39 dimensions. Models are trained in a maximum likelihood way on the training samples as described in Section 5. The Gaussian mixtures are grown incrementally over several iterations.

Since the GMM based decoder classifies speech segments on a per frame basis and only uses a one frame context from the delta and double delta features, speech segments are cut off very tightly. The speech recognition system can handle non-speech frames that were misclassified as speech, but false negative frames can not be recovered. Expanding the speech segments on both sides by 0.4 seconds improved the segmenter's performance.

4 SVM Based Method

SVMs have already been applied to the closely related voice activity detection (VAD) problem in the past by [8]. Since then, several works such as [9] extended these ideas. The latter work, among many others defines the speech/non-speech discrimination problem as two-class discrimination problem. With works like [10] there also exist studies that extend this task to a multi-class problem by splitting speech/non-speech detection into sub-tasks. Similar to [11], our main objective is to maximize the improved intelligibility of speech under noisy channel conditions.

As SVMs naturally model two-class decision cases, it is straightforward to train a model on reference samples mapped to two distinct classes for speech and non-speech. The mapping is performed as described in Section 5. However, no exact phoneme-to-audio alignments that could serve as references were accessible for our experiments, thus it has been decided to perform training on previously computed labels that have been generated by our baseline system. Consequently, the references for training are not exempt from errors, albeit the quality of references still being high enough to enable an effective training.

4.1 SVM Training

The classifier is trained on the $(train_{svm})$ set, using the LIBSVM library [12]. We decided to use the C-Support Vector Classification (C-SVC) formulation, as it is the original SVM formulation [13], and fits our requirements. The SVM will find a hyperplane a high-dimensional space, which separates the the classes in a linear fashion and with a maximal margin between them. With a soft margin parameter $C > 0$, a penalty parameter of the error term can be used for adjustment [12]. The decision function we use for classification is:

$$\operatorname{sgn}\left(\sum_{i=1}^{l} \boldsymbol{y}_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \quad \text{with } K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2} \tag{1}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the RBF kernel function [12]. The values for (C, γ) are determined automatically during classifier training via a steepest ascent hill climbing algorithm by optimizing the frame based classification accuracy on (dev_{svm}). The C and γ are exponentially growing, following the recommendation of [14]. In order to avoid numerical problems, vector scaling is applied during training and testing [14].

Feature Selection. Initial experiments aimed at the identification of the most useful front-end for automatic segmentation. Similar to [15], a major focus was on testing standard feature vectors such as MFCCs, as they are commonly used for solving other automatic speech processing tasks. Loosely related to [16], we also utilize linear discriminant analysis (LDA) for preferably low-loss dimensional reduction.

The following front-ends have been evaluated: a) standard logMel feature vectors comprising 30 parameters b) standard 13 dimensional MFCC feature vectors c) 15 adjacent MFCC vectors stacked and LDA-transformed.

Our experimental evaluations on dev_{svm} show that it is always of advantage to integrate temporal information. In all cases, the SVMs trained on stacked variants of feature vectors outperformed the non-stacked variants. Moreover, stacking 15 adjacent frames outperformed the computation of Δ and $\Delta\Delta$. Ultimately, the systems using LDA-transformed feature vectors outperformed all other alternatives.

Further improvements were obtained by adding various features such as frame based peak distance or zero crossing rate. The enhancements were tested with all front-ends but the LDA based vectors. Where stacking or Δ computation was applied, the feature vectors were extended before the respective operation.

In order to minimize the dimensionality for reduced training complexity, we experimented with feature selection via the *f-score* measure. Except for logMel feature vectors, the discriminative capabilities of the original features are higher for low dimensions and gradually decrease for higher dimensions. None of the solutions with lower dimensionality was able to outperform the original systems. Contrary to expectations the discriminative abilities of the resulting models decreased, thus rendering dimensional reduction inefficient.

4.2 Post-processing

The output of SVM classification is a string of 1's and 0's, hypothesizing whether a frame belongs to a speech or non-speech region. However, operating on frame-level is too fine-grained and makes post-processing necessary to obtain a useful segmentation of the audio. For smoothing the raw SVM classification data we follow a 2-phase approach. First, smoothing is performed on frame-level to remove false positives and false negatives. Then, merging on segment level is performed to acquire more natural segments of reasonable length.

Smoothing on Frame-Level. The smoothing technique we apply is derived from the well-known *opening* strategy of morphological noise removal in computer vision. An *erosion* step on sequences of frames hypothesized as speech is followed by a *dilation* step. A major difference to the classic operation is that our algorithm is extensive, i.e, the resulting segment is larger than the original. This is achieved by differing factors for erosion and dilation, where $f_{erode} < f_{dilate}$. As a result, the idempotence property of the opening algorithm is also lost.

Our intention is to remove very short segments classified as speech by setting $f_{erode} = 3$, under the assumption that these are likely to be noise or artifacts arising from channel characteristics. A factor of 3 leads to a cut-off of 36 milliseconds of audio on each side of the respective hypothesized speech segment, and the deletion of isolated segments with a duration below 72 milliseconds. These values roughly approximate the estimates for minimal and average phoneme lengths [17]. By a stronger dilation, short gaps of predicted non-speech between parts of speech shall be removed. This has several justifications: For one, the dilation has to compensate for the erosion operation. Then, we follow the assumption that it is likely to reduce falsely rejected speech parts by closing comparatively short gaps. Furthermore, lost data by erroneous segmentation is more harmful than the inclusion of potentially noisy parts for decoding. To avoid too strict segment borders, the dilation step further serves as *padding* operator, extending the segment borders to a certain degree.

Segmentation Generation. Commonly, automatic segmentation maintains a minimal distance seg_{dist} between individual segments, e.g., for establishing sentence-like structures. Our goal was to exclude especially large parts of silence from decoding, and to minimize the occurrence of artifacts, without loss of relevant information. Both phenomenons directly arise from the nature of the recorded telephone conversations. A minimal distance between speech segments was defined by setting $seg_{dist} = 0.5$ milliseconds. Segments with a lower gap in between are merged. Moreover, isolated parts in the signal hypothesized as speech, but having a very short duration are pruned away.

5 Data Selection

To get the training samples for the speech and non-speech models we used forced alignments of the training data with the provided references. For alignment we used the same system than for the decoding experiments, or at least one of similar performance.

For the two-class case of speech/non-speech classification, a phoneme mapping was defined that maps phonemes modelling linguistic sound units to a speech category, and models that represent phenomenons that are considered noise and filler entities to a non-speech category. In the GMM-framework, additionally, non-speech samples classified as silence are mapped to a silence category.

We developed our systems for the Tagalog *unlimited* training data condition, that means around 100 hours of transcribed audio training data was available.

For the GMM based decoder method computational resources were no critical issue for the model training, so all data was used for training.

For the SVM based approach, the vast amount of training samples renders a training on the full data set entirely infeasible. Thus, a sample subset of approx. 200.000 samples was selected as training set ($train_{svm}$), and approx. 100.000 samples were used as development test set (dev_{svm}). Data extraction was conducted equally distributed among the target classes. Therefore, sample vectors were extracted phone-wise. From each utterance, an equal amount of samples was extracted to cover all data. Further, the extraction considers the shares of phonemes in the data. Every sample is belonging to either speech or non-speech, according to the pre-defined mapping. This way, both classes see an equal amount of data, equally distributed over the full data set and representing each original phoneme class according to their respective proportion.

6 Experiments

As ground truth for our experimental evaluation we used the manually generated transcriptions that came along with the development data. It is to distinct between two conditions: First, we performed the experimental evaluation on a test set of the same language the development data belongs to. In addition to this test series, four more automatic segmentation systems were trained for each proposed approach, each in another distinct language, where three of the languages are new and previously unseen during development. Thus, the optimized training pipelines are straightforwardly applied to the new conditions, allowing for evaluation of the generalization capabilities of our setups.

Table 1. Results for the Tagalog unlimited training data condition

	WER	Subst.	Del.	Ins.	#Seg.	dur.	avg.	max.
manual	63.1%	39.3%	16.9%	6.9%	11353	10.7h	3.4s	35.5s
baseline	62.6%	39.5%	15.6%	7.5%	12986	11.1h	3.1s	30.0s
GMM-based	61.9%	37.6%	18.5%	5.9%	15188	9.7h	2.3s	29.3s
SVM-based	62.4%	38.8%	16.5%	7.2%	15293	8.8h	2.0s	36.4s

Table 1 shows that both automatic segmentation approaches can outperform the manual segmentation for *Tagalog*. Our segmentations are further compared to the baseline for automatic segmentation (see 2). The segmentations of both approaches lead to a decrease in WER, if compared to the baseline.

Further analysis reveals considerable differences in the nature of the individual segmentations. By reference to Table 1 it can be seen that the amount of automatically determined segmentations is considerably higher, with at the same time notably shorter average segment length. The higher degree of fragmentation of the audio data leads to a lower accumulated duration. At the same time, recognition accuracy is not only maintained, yet even increased.

Table 2 lists the evaluation results for the *limited* case on all four languages. A direct comparison between both training data conditions of Tagalog reveals that the GMM-based segmentation proves to be superior when applied on the full training set, but it is the SVM-based segmentation that wins over the alternative, when having only a limited amount of training data at hand. For the other languages, none of the automatically generated segmentations can outweigh the manual partition. In the cases of *Cantonese* and *Turkish*, however, the difference to the performance on manual segmentations is

Table 2. Results in WER for the limited training data conditions on all four languages

Segmentation	Tagalog	Cantonese	Pashto	Turkish
manual	78.5%	76.7%	77.5%	74.0%
GMM-based	77.5%	76.8%	78.5%	74.5%
SVM-based	76.9%	76.9%	78.4%	74.3%

0.67% relative at the most. For *Pashto*, both automatic approaches are not able to reach the accuracy of the manual generated data, with 1.2% relative difference to the latter.

7 Conclusion and Future Work

This paper compares model based segmentation methods for unsegmented telephone conversations. Two methods based on the use of general speech and non-speech models (one GMM based and one SVM based method) are compared to a standard method that uses a general speech recognition system. We showed that our speech/non-speech modelling based segmentation methods achieve comparable results to those of manual segmentation. For larger amounts of training data, the GMM based method performed best, while the SVM based method is preferable if the amount is limited.

The languages we worked on are low resourced and not as well investigated as other languages and the corresponding systems we used achieve high WERs. Since the purity of the training data for the models for segmentation depends on the quality of the alignment and therefore on the speech recognition system, the methods have to be evaluated on well researched languages. Moreover, the dependency on the amount of training data could be investigated.

For the GMM based method there are several parameters that have to be optimized. The use of bottle-neck features improves speech recognition significantly (e.g. [18]) so the application on the segmentation seems to be promising. Increasing the front-end's window size should be investigated in general.

LIBSVM provides probabilistic classification, which might be topic of further experiments on SVM-based segmentation. Besides LDA, other transformations could be utilized for reduction of dimensionality. Within the scope of this research, the effect of additional features before LDA transformation remained open.

Acknowledgements. Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

References

1. IARPA: IARPA, Office for Incisive Analysis, Babel Program, `http://www.iarpa.gov/Programs/ia/Babel/babel.html` (retrieved March 06, 2013)
2. Stüker, S., Fügen, C., Kraft, F., Wölfel, M.: The ISL 2007 English Speech Transcription System for European Parliament Speeches. In: Proceedings of the 10th European Conference on Speech Communication and Technology (INTERSPEECH 2007), Antwerp, Belgium, pp. 2609–2612 (August 2007)
3. Soltau, H., Metze, F., Fügen, C., Waibel, A.: A One-pass Decoder Based on Polymorphic Linguistic Context Assignment. In: ASRU (2001)
4. Yu, H., Tam, Y.C., Schaaf, T., Stüker, S., Jin, Q., Noamany, M., Schultz, T.: The ISL RT04 Mandarin Broadcast News Evaluation System. In: EARS Rich Transcription Workshop (2004)
5. Kraft, F., Malkin, R., Schaaf, T., Waibel, A.: Temporal ICA for Classification of Acoustic Events in a Kitchen Environment. In: INTERSPEECH, Lisbon, Portugal (2005)
6. Beyerlein, P., Aubert, X., Haeb-Umbach, R., Harris, M., Klakow, D., Wendemuth, A., Molau, S., Ney, H., Pitz, M., Sixtus, A.: Large Vocabulary Continuous Speech Recognition of Broadcast News - The Philips/RWTH Approach. Speech Communication 37(12), 109–131 (2002)
7. Matsoukas, S., Gauvain, J., Adda, G., Colthurst, T., Kao, C.L., Kimball, O., Lamel, L., Lefevre, F., Ma, J., Makhoul, J., Nguyen, L., Prasad, R., Schwartz, R., Schwenk, H., Xiang, B.: Advances in Transcription of Broadcast News and Conversational Telephone Speech Within the Combined EARS BBN/LIMSI System. IEEE Transactions on Audio, Speech, and Language Processing 14(5), 1541–1556 (2006)
8. Enqing, D., Guizhong, L., Yatong, Z., Xiaodi, Z.: Applying Support Vector Machines to Voice Activity Detection. In: 2002 6th International Conference on Signal Processing, vol. 2, pp. 1124–1127 (2002)
9. Ramirez, J., Yelamos, P., Gorriz, J., Segura, J.: SVM-based Speech Endpoint Detection Using Contextual Speech Features. Electronics Letters 42(7), 426–428 (2006)
10. Lopes, C., Perdigao, F.: Speech Event Detection Using SVM and NMD. In: 9th International Symposium on Signal Processing and Its Applications, ISSPA 2007, pp. 1–4 (2007)
11. Han, K., Wang, D.: An SVM Based Classification Approach to Speech Separation. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4632–4635 (2011)
12. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
13. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20, 273–297 (1995)
14. Hsu, C.W., Chang, C.C., Lin, C.J.: A Practical Guide to Support Vector Classification (2010)
15. Kinnunen, T., Chernenko, E., Tuononen, M., Fränti, P., Li, H.: Voice Activity Detection Using MFCC Features and Support Vector Machine (2007)
16. Temko, A., Macho, D., Nadeu, C.: Enhanced SVM Training for Robust Speech Activity Detection. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007, vol. 4, pp. IV–1025–IV–1028 (2007)
17. Rogina, I.: Sprachliche Mensch-Maschine-Kommunikation (2005)
18. Kilgour, K., Saam, C., Mohr, C., Stüker, S., Waibel, A.: The 2011 KIT Quaero Speech-to-text System for Spanish. In: IWSLT 2011, pp. 199–205 (2011)