DISCRIMINATIVE TRAINING OF MULTI-STATE BARGE-IN MODELS

Andrej Ljolje, Vincent Goffin

AT&T Labs - Research
Florham Park, NJ 07932-0971
U.S.A.
{alj,vjg}@research.att.com

ABSTRACT

A barge-in system designed to reflect the design of the acoustic model used in commercial applications has been built and evaluated. It uses standard hidden Markov model structures, cepstral features and multiple hidden Markov models for both the speech and non-speech parts of the model. It is tested on a large number of real-world databases using noisy speech onset positions which were determined by forced alignment of lexical transcriptions with the recognition model. The ML trained model achieves low false rejection rates at the expense of high false acceptance rates. The discriminative training using the modified algorithm based on the maximum mutual information criterion reduces the false acceptance rates by a half, while preserving the low false rejection rates. Combining an energy based voice activity detector with the hidden Markov model based barge-in models achieves the best performance.

Index Terms: barge-in, VAD, dialog systems, speech recognition, acoustic modeling

1. INTRODUCTION

Speech processing technologies have since their inception been involved with the problem of detecting speech, whatever the acoustic environment. The problem of accurately distinguishing speech from the background is still an active area of research as it is so important, and so hard to do well.

In practice there are three different applications involving speech detection. They differ in their intent and the mechanisms used to achieve their targets. The most obvious is the general question of the presence of speech, commonly referred to as the Voice Activity Detection (VAD) problem, which tries to detect every non-speech segment, even if it is, for example, within a continuous utterance, like a short pause [1, 2, 3]. The next obvious application, most commonly encountered in ASR applications is the problem of endpointing. Here we only need to detect the beginning and the end of an utterance, but we rely on the ASR system to internally determine if there are any utterance internal pauses [4, 5, 6]. Of course, they are still detected, and if long enough, based on some pre-determined threshold, an and of an utterance is declared.

Here we address the problem of a unique speech detection problem that only occurs in dialog based applications: barge-in. Barge-in happens when a user of an automated dialog system attempts to input speech during the playback/synthesis of a prompt generated by the dialog. In this unique situation two things are expected to occur, virtually instantaneously. First the prompt is immediately terminated, both to indicate to the user that the system is listening to him/her, and to allow uninterrupted recognition of the user's utterance. At the same time, the ASR engine starts processing the accumulated speech starting some short amount of time prior to the detected barge-in [7]. In the case of barge-in we face only a relatively small subset of the problems faced by the VAD systems. On the other hand, the errors can have a significant impact to the perceived usability of the system and might cause it to be abandoned by the user. A false barge-in, which happens when the system incorrectly believes that there is speech input by the user will terminate the prompt, leaving the user without proper guidance for providing the appropriate input to the system. This can have a long term effect diverting the dialog away from the intended operation for many turns. On the other hand, if, by trying to minimize false alarms, the system becomes less sensitive to speech input and fails to barge-in, the users find it uncomfortable speaking while the prompt is still active, which corrupts their delivery of the speech input, affecting the ASR due to the unnaturalness of the input. In addition this often leads to unwanted echo and consequent poor recognition performance. This is assuming the ASR system is left active all the time, and not initiated by the barge-in detection, in which case the speech would be lost to the system.

The ideal barge-in response requires minimum latency, responding to the speech input as quickly as possible, while requiring high level of accuracy in detecting speech. Those two criteria are contradictory and are often traded off one against the other. The overall dialog system scenario implies, to a large extent, that the barge-in performance is tightly coupled with the ASR system. In essence, a flawless barge-in performance that negatively impacts the ASR performance is detrimental to the system performance, and vice-versa. In many ways the best barge-in system is the ASR system, with the serious drawback that its latency is too long. Alternatively, matching the barge-in performance to the ASR performance would minimize such possible differences and it is naturally achieved by using the ASR technology to provide the barge-in processing.

Here we attempt to discriminatively train a hidden Markov model (HMM) based barge-in acoustic model. Unlike the more conventional Gaussian mixture models (GMMs) which normally have one GMM for speech and one for non-speech, which leads to easy input labeling, multi-state HMMs have a large number of correct potential alignments. All alignments which satisfy the condition that some of the HMM states (the speech states) are aligned with the speech portion of the signal and some of the non-speech states are aligned with the rest of the input. Since discriminative training techniques require the "truth" to be known this ambiguity creates a difficulty. Here we show an easy method for avoiding this obstacle resulting in greatly improved barge-in performance on a large number of speech utterances from different applications.

2. THE SPEECH DATA

One of the difficulties of training and evaluating barge-in models is the inability to replicate the real-life barge-in conditions in sufficient numbers for collecting truly representative data. Even when this is ignored, it is necessary to label a large amount of data for speech and non-speech which can be very time consuming, especially if done on large amounts of speech. It is necessary, however, to evaluate on large databases to achieve an accurate representation of barge-in performance on different tasks under realistic conditions. We circumvent all of these problems by using all of the data which was collected for training the latest version of the AT&T acoustic model for commercial applications [8, 9]. This database of well over 1000 hours of speech, over a million utterances and about 10 million words consists of speech collected as part of dozens of different collection scenarios, from real life customer applications to recordings designed with specific targets for data collection. It includes general English utterances, alphabet and digit strings, both in isolation and combined, isolated utterances, short utterances like names, and many others. It provides a great variability for both training the barge-in models and testing of barge-in performance. Unlike most publications on speech detection, we are not so concerned with artificially adding noise to clean speech recordings to simulate what might happen in real-life conditions. We already have the appropriate data, and our task is to evaluate and improve the performance on the data our recognition system encounters in its usual applications. Given the size of the data, we concluded to approach the barge-in problem from a different perspective. First, the models we were planning to build are so small relative to the amount of data we had, that we could ignore the question of overtraining. Consequently, in addition to several small independent test sets, our training data is also used as out testing data. The main problem was the labeling of so much speech, but given the quantity we decided that even noisy labeling was going to be adequate. We used the current version of the acoustic model trained on this data for forced alignment of the speech with the lexical transcriptions. For our convenience, the speech segments were excised and became the training data for the speech part of the barge-in model. Similarly, the non-speech parts of the database were excised to become the training data for the non-speech part of the barge-in model. Thus it is our ground truth for speech/non-speech boundaries.

3. THE BARGE-IN MODEL

The barge-in model we chose to investigate reflects our experience in building acoustic models for ASR and the needs and capabilities of the recognizer for which it is built. Matching the ASR and the barge-in configurations as much as possible seemed optimal, using multiple HMMs for speech and non-speech. Also, given our past experience with modeling non-speech portions of the signal for maximum recognition accuracy, a single GMM is inadequate. Consequently, we chose to preserve the non-speech part of the acoustic model [8] in the barge-in model. It consists of four HMMs, two with a single state, and two as three state left-to-right HMMs, totaling eight states, each represented as a Gaussian mixture. Each weighted mixture had 32 components, each having 13 dimension corresponding to 13 cepstral features. The decision to preserve the non-speech part of the ASR model made the bootstrapping trivial, as we could just keep the original HMM segmentations and labels. For the speech part of the model, we continued with keeping the characteristics of the ASR model. We partitioned the phoneme labels in the ASR model into five categories: vowels and glides, unvoiced fricatives, voiced fricatives, other consonants, and nasals. The speech part of the barge-in model consists of five three state left-to-right HMMs, based on the ASR model phonemic segmentation of the training data, after relabeling into one of the five phoneme classes.

The training data is processed using a standard 13-dimensional mel-filterbank cepstral analysis every 10 ms. No additional processing was used to minimize latency and processor usage. The training of the barge-in model followed the usual steps of training a recognition acoustic model. We preserved the speech/non-speech boundaries, but allowed for new legal HMM sequences within the speech and non-speech segments as part of the training. We forced an arbitrary sequence of non-speech HMMs for the non-speech segments. Initially, for the speech segments, in addition to forcing the use of the speech HMMs, non-speech HMMs were also permitted, but with a very high insertion cost. It can be thought of as a language model cost, which was set to 6 for insertion of a non-speech HMM, with the language model weight of 16. All other HMMs have the insertion cost set to 1. As will be seen later, the use of insertion cost can be used to manipulate the trade-off between the false insertion and false acceptance in the barge-in performance. The reason for allowing non-speech HMMs during the speech segments is that automatic forced alignment of recognition acoustic models, due to their context-dependent HMM structure and thus somewhat arbitrary placement of phoneme boundaries, often exhibit the tendency to include some of the non-speech portion of the signal as part of the utterance initial or final phoneme. Each HMM had a gamma duration distribution associated with it, and the weight given to the duration model was the same as the weight given to the language model. The maximum likelihood (ML) training of the barge-in model which consisted of several iterations of Viterbi training on all the available data produced the initial performance reference, expressed, as all the other results on the training data here, as an ROC curve between false acceptance (detecting speech during non-speech segments) and false rejection (failing to detect speech when present). The model performance is adjusted using the very simple logic of detecting contiguous speech segments of n frames, with $n = \{1, 5, 10, 15, 20, 25, 30\}$. If such a segment is detected within the first 350 ms, or 35 frames, than it is considered a correct detection. If it is detected too late it is a false rejection, and if a speech segment is detected anywhere during the non-speech segments, it is considered a false acceptance.

4. DISCRIMINATIVE TRAINING

The discriminative training of the barge-in model is based on the Maximum Mutual Information criterion (MMI) [10]. We follow the implementation as described in [11], except that our implementation is based on the Viterbi alignments of the correct word sequence (one best), rather than a lattice of the possible paths, given the reference transcription for each of the utterances.

The problem of barge-in model training can be viewed as a speech recognition training scenario. It can be represented in two different ways. First, we can view it as a two word problem, with speech being one word, and non-speech as the other. The different HMMs can be viewed as the phoneme inventory, and any phoneme sequence is a valid alternative pronunciation, as long as only speech HMMs are used for the speech "word", and non-speech HMMs are used for the non-speech "word". The other approach is to think of the HMMs as the words, where any word sequence of the speech "words" during the speech segment is valid, and similarly any sequence of non-speech "words" is valid during the non-speech segment. In practice, this makes little difference as the training process

ends up doing the same steps.

In our implementation, we viewed HMMs as words. The inherent problem in training multi-HMM barge-in models is that there are many valid HMM/word sequences that satisfy the reference sequence criterion. Thus it is impossible to define what the reference transcription is, as it is based on the model being trained. The reference is defined by using the model to force and alignment with a valid set of alternatives in the network. As the training progresses, the model changes and the reference sequence changes with it. To alleviate this problem, the reference transcription forced alignment is replaced with a restricted grammar recognition. The hypothesis lattice is generated by allowing any sequence of the speech and non-speech HMMs. The reference lattice is obtained by recognizing the most likely HMM sequence, but allowing only speech HMMs for the speech segments, and non-speech HMMs for the non-speech segments. This way, different iterations of the ML and MMI training end up having different reference transcriptions. However, the restriction that the speech segments are only matched with the speech HMMs and the non-speech segments are matched with the non-speech HMMs is preserved. This is relaxed, slightly, to allow for the non-speech HMM alignments during the possible mis-labeling at the beginning and end of the speech segments as described earlier, since the manual transcriptions in terms of speech and non-speech were not available.

5. EXPERIMENTAL RESULTS

5.1. Testing on the Training Data

The test data used in these experiments is the same as the training data, consisting of over a million utterances. As in the training, it has been partitioned into speech and non-speech segments, as determined by forced alignment of the lexical reference transcriptions.

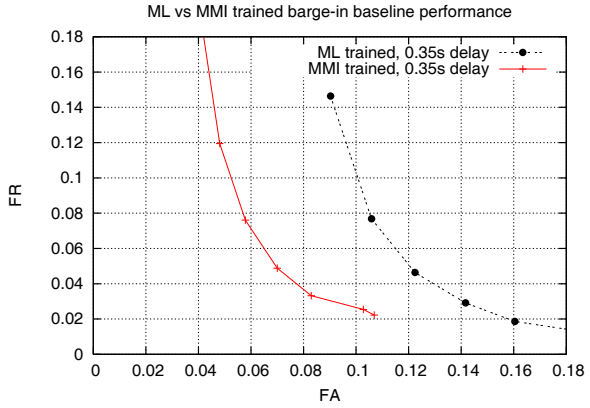

Fig. 1. Baseline performance with the ML and MMI trained barge-in models, trained and tested on all available data

The initial experiments utilized all the segments in training and evaluating the model. In that respect it followed the VAD scenario rather then the barge-in requirements. The barge-in performance of the ML trained and the MMI trained model is shown in Figure 1. In order for the barge-in performance to be considered acceptable, the speech detection had to occur within the first 350 ms (35 frames).

It clearly demonstrates the benefit of discriminative training, and the trade off between false acceptance (FA) and false rejection (FR) performance. The low FR end of the curve corresponds to detecting a single speech frame by the decoding of the barge-in model. The low

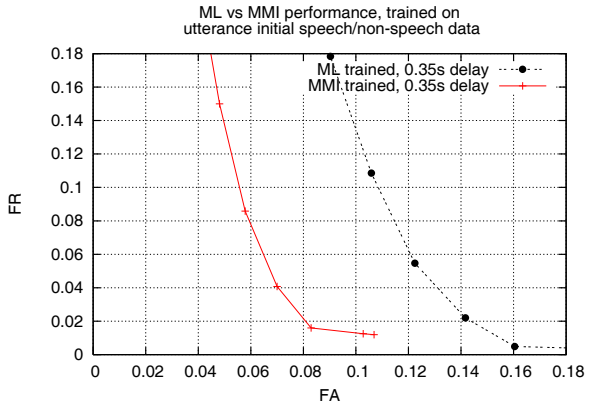

Fig. 2. Performance with the ML and MMI trained barge-in model, trained and tested only on the initial non-speech and speech data

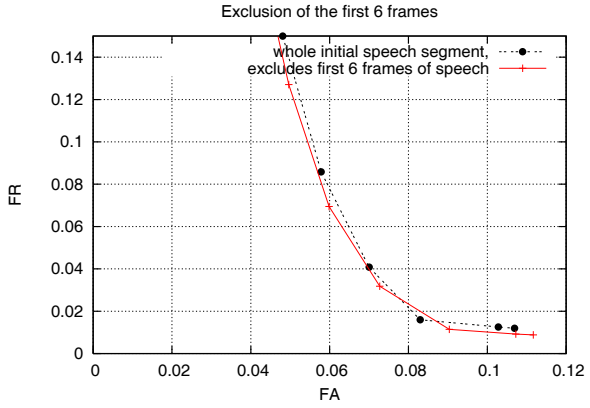

Fig. 3. Performance with the MMI trained barge-in model when trained on whole speech segments and with the first 6 frames removed

FA end is achieved by detecting the minimum of $n = 30$ frames, both within the first 350 ms of the speech segment, or within the segment, regardless of length. In reality, most of the detections occur much before the 350 ms are up, and many initial speech segments are shorter than 350 ms, making the FR score at the high FR end of the curve look worse than they are. However, close to the operating point of 10-15 frames limit, the performance is depicted accurately.

Given that the intended use for the model was barge-in, the next configuration used only the initial silence preceding the utterance to train the non-speech HMMs, and only the initial speech segment, before any pauses and only up to 50 frames (0.5 s in length). The performance is shown in Figure 2.

The benefit of this approach is mostly reflected in much reduced FR performance.

We next attempted to remove the problem of moderately frequent addition of a few frames of non-speech to the beginning of the initial speech segments. Also, few utterances appeared to be erroneous, as the initial speech segments had the length of only a few frames. In the next results, the first 6 frames of the speech segment were removed from the training and testing speech segments. Also, all the speech segments of less than 15 frames were discarded. Given that the first 6 frames were removed, the minimum segment length was 9 frames. Fortunately, so few utterances were discarded due to the short length that this change did not effect the ROC curves. The

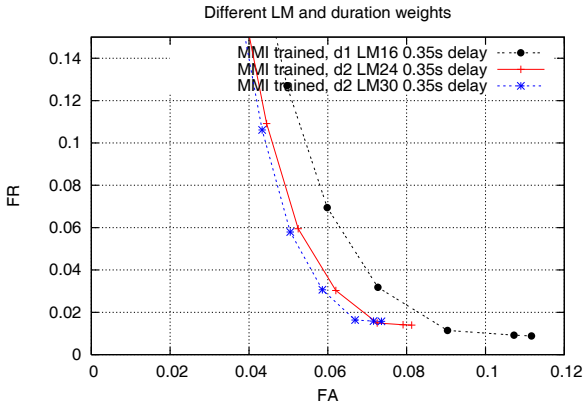

Fig. 4. Performance MMI trained barge-in model, trained and tested only on the initial non-speech and speech data, with different language and duration model weights

comparison of the performance by the MMI trained models before and after these two changes is shown in Figure 3.

The improvement is at best modest and it appears that it is not necessary to provide special handling for the infrequent inaccuracies of the segmentation into speech/non-speech segments by forced alignment.

The final experiment compares the effect of varying the cost of inserting speech and non-speech HMMs. Given that the search network already had costs associated with inserting any of the nine HMMs, this is easily achieved by changing the network (language model) cost weight when doing the decoding. The same can be done with the duration model as well. We compare three different settings in Figure 4.

The small loss in FR performance is more than offset by the improvement in the FA performance as the LM and the duration weights are increased.

5.2. Testing on Previously Unseen Test Data

Once the barge-in models were integrated into the AT&T Watson recognition engine the performance was evaluated on a number of diverse tasks, using recordings that were not used in creating the barge-in models. The integrated barge-in system consists of only the original energy-based VAD, or the combined energy based VAD and the HMM based barge-in detector, with the HMMs trained using the ML or the MMI criterion. The energy-based VAD [12] is based on noise threshold adaptation for voice activity detection in nonstationary noise environments. This is particularly useful as many hardware platforms for dialog processing allow for too much prompt echo to leak into the speaker channel, which would regularly trigger false barge-ins without the noise threshold adaptation. Also, more significantly, the same problem plagues purely HMM based barge-in. The simple solution of an "AND" function on the decisions from the energy based VAD and the HMM-based barge-in system simply removes the problem, and ultimately provides the best overall performance. Additional optimization of the interaction between the different barge-in components is likely to provide further modest improvements to the performance described here. The combining of the energy-based VAD and the ML trained HMMs significantly improves the performance of the HMM barge-in detector, and modestly improves the energy-based VAD as a barge-in detector. On the other hand, after MMI training, combining the two detectors marginally improves the performance of the HMM detector alone, but significantly improves the performance of the VAD when it used in isolation.

Unlike the experiments on the training data, where the only parameter was the number of consequtive frames labeled as speech that were necessary to declare a barge-in event, the Watson implemtation defines a sensitivity parameter, which combines two metrics: the likelihood difference between the speech and non-speech states, and the length over which the difference in likelihood is accumulated. The performance details are best observed in a form of a histogram, with the vertical green line indicating the speech start position, and the histogram indicates counts of detections at a particular frame in the speech signal. The detection in general occurs 20-30 frames after the speech starts. If the detection occurs before the speech started, it clearly indicates a false barge-in event, and if it occurs more than 0.5 secs after the speech started (an arbitrarily chosen point) it is declared to be a false rejection of a spech event. Examples of barge-in histograms are shown in Figure 5.

The best way to interpret the performance of the barge-in system described here is to compare the false barge-in (too early) false rejection (too late) or both added up as a percentage of the total number of utterances relative to the barge-in sensitivity setting. The lower the setting the lower the latency, however low latency should not be achieved at the expense of accuracy. Figure 5 shows the histograms with the sensitivity set to 50 (mid-point of the sensitivity scale). Figures 6, 7, 8, 9, 10 and 11 show the performance for the VAD alone, VAD + ML HMMs and VAD + MMI HMMs respectively on six different tasks. The performance advantage of the MMI trained HMM based approach is immediately obvious and is most significant at low sensitivities which provide shortest latencies and best user experience during a barge-in event.

6. CONCLUSIONS

A barge-in system has been developed for use in dialog systems, attempting to maximize the performance of the complete system. In order to achieve this goal the barge-in model was designed to in many ways mimic the ASR model, including using the same non-speech HMMs, and similar number of HMMs to represent the speech segments. The training followed the standard ML and MMI training steps, except that the reference transcriptions were redefined at every iteration of training, since they are arbitrary for speech/non-speech determination. The HMM based barge-in model can achieve very low FR rates, and the MMI training reduces the FA acceptance rate by a half at a given FR operating point. Given how often barge-in problems affect the dialog system performance [7], such a large performance improvement in false acceptance rate implies a significant improvement in dialog completion rates and customer satisfaction scores.

7. REFERENCES

[1] de la Torre, A., Ramirez, J., Benitez, C., Segura, J., Garcia, L. and Rubio, A., "Noise robust model-based Voice Activity Detection", In *Proceedings ICSLP,* pp. 1954-1957, Pittsburgh, PA, 2006.

[2] Cournapeau, D., Kawahara, T., Mase, K. and Toriyama, T., "Voice Activity Detector Based on Enhanced Cumulant of LPC Residual and On-line EM Algorithm", In *Proceedings ICSLP,* pp. 1201-1204, Pittsburgh, PA, 2006.

[3] Ishizuka, K. and Kato, H., " A Feature for Voice Activity Detection Derived from Speech Analysis ith the i Exponential Au-

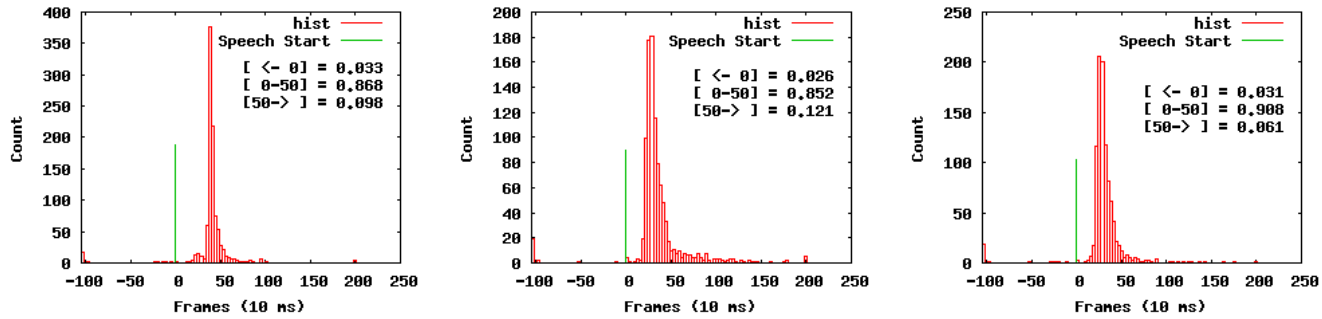

Fig. 5. Histogram of barge-in event detection relative to the true speech start time on an alpha-digits task using just the VAD, VAD + ML HMMs and VAD + MMI HMMs

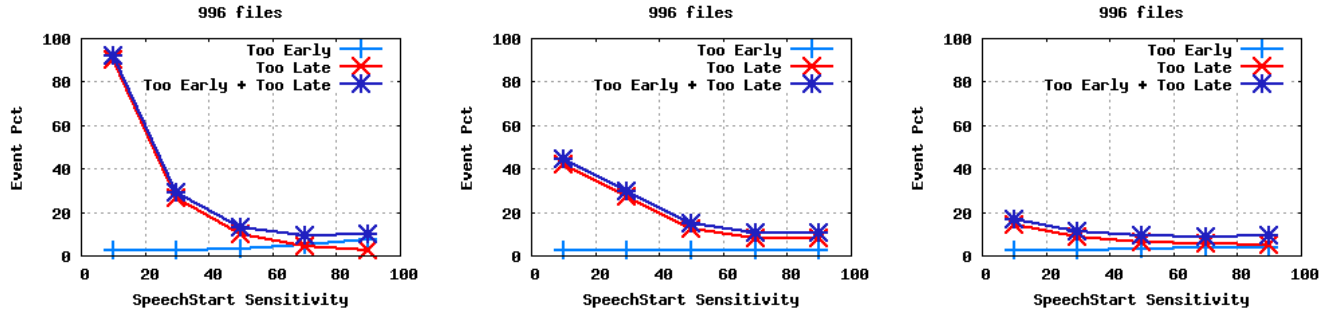

Fig. 6. Barge-in performance on an alphadigits task: VAD alone, VAD + ML HMMs and VAD + MMI HMMs

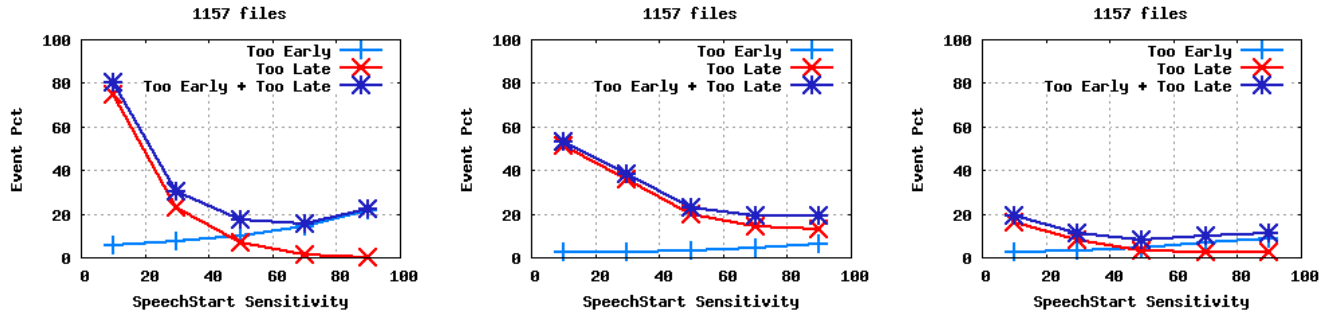

Fig. 7. Barge-in performance on a directed dialog task: VAD alone, VAD + ML HMMs and VAD + MMI HMMs

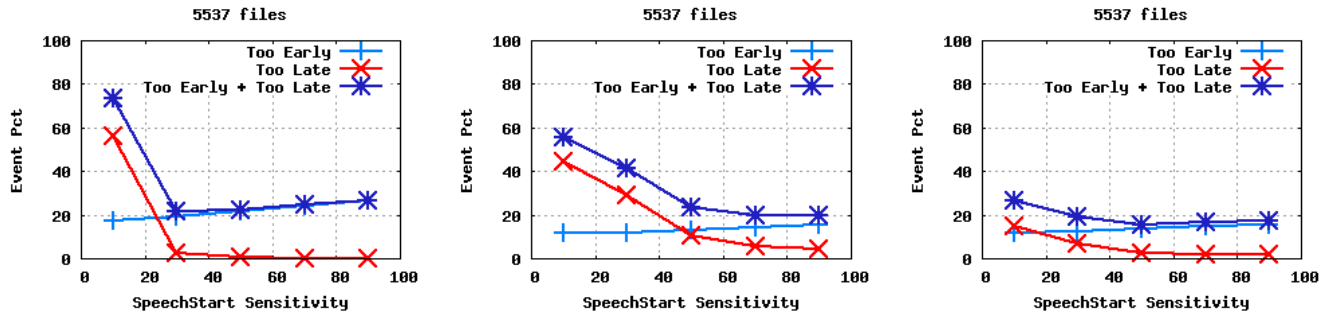

Fig. 8. Barge-in performance to a "How may I help you?" prompt response: VAD alone, VAD + ML HMMs and VAD + MMI HMMs

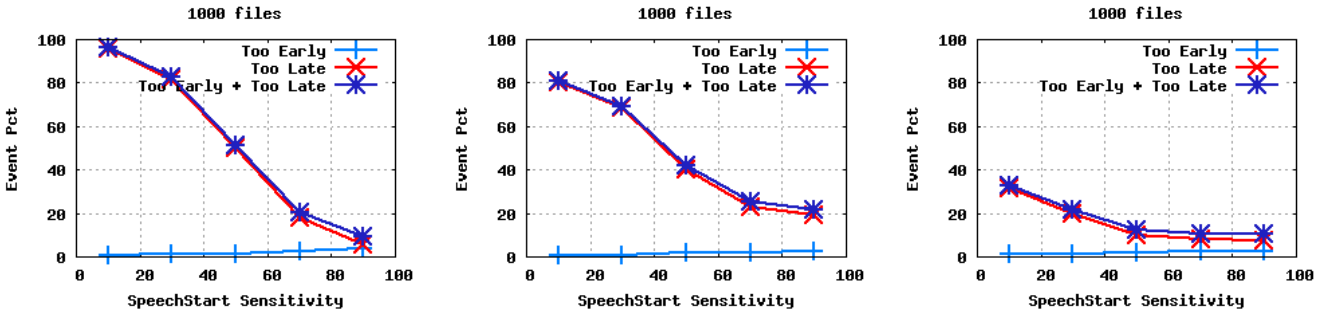

Fig. 9. Barge-in performance on a confirmations (yes/no) task: VAD alone, VAD + ML HMMs and VAD + MMI HMMs

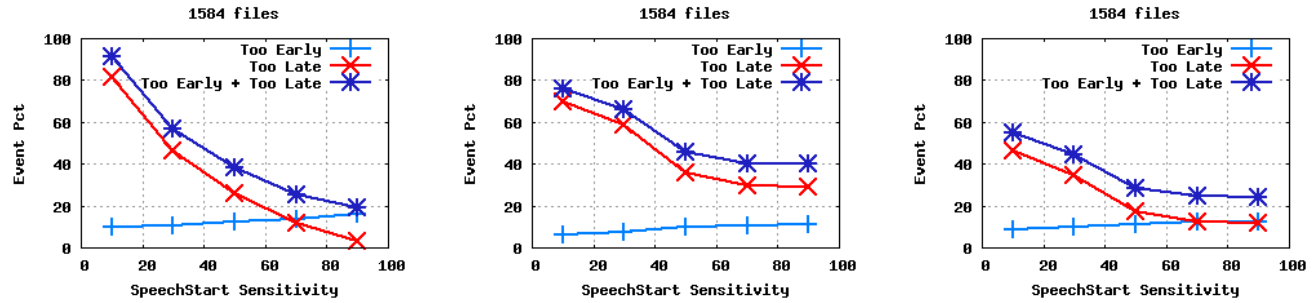

Fig. 10. Barge-in performance on a digits task: VAD alone, VAD + ML HMMs and VAD + MMI HMMs

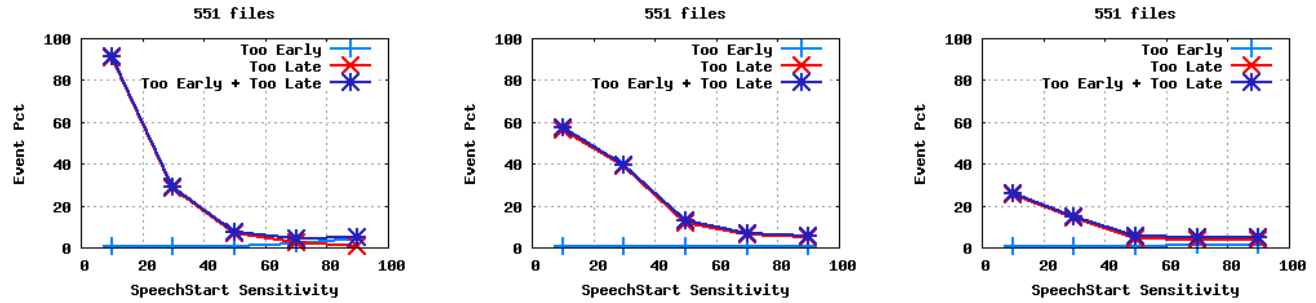

Fig. 11. Barge-in performance on a names recognition task: VAD alone, VAD + ML HMMs and VAD + MMI HMMs

toregressive Model", In *Proceedings ICASSP,* pp. I–789-I–792, Toulouse, France, 2006.

[4] Huang, L.-S. and Yang. C.-H., "A Novel Approach to Robust Speech Endpoint Detection in Car Environments", In *Proceedings ICASSP,* pp. 237-240, 2001.

[5] Li,, Q., Zheng, J., Tsai, A. and Zhou, Q., "Robust Endpoint Detection and Energy Normalization for Real-Time Speech and Speaker Recognition", *IEEE Transactions on Speech and Audio Processing,* Vol. 10, No. 3, pp. 146-157, March 2002.

[6] Yamamoto, K., Jablun, F., Reinhard, K. and Kawamura, A., " Robust Endpoint Detection for Speech Recognition Based on Discriminative Feature Extraction", In *Proceedings ICASSP,* pp. I–805 - I–808, Toulouse, France, 2006.

[7] Raux, A., Bohus, D., Langner, B., Black, A. and Eskenazi, M., "Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience," In *Proceedings ICSLP,* 2006.

[8] Ljolje, A., "Multiple task-domain acoustic models," In *Proceedings ICASSP,* 2003.

[9] Ljolje, A., "Optimization of Class Weights for LDA Feature Transformations," In *Proceedings ICSLP,* 2006.

[10] Bahl, L., Brown, P., de Souza, P. and Mercer, R., "Maximum Mutual Information Estimation of Hidden Markov Model Parametersfor Speech Recognition," In *Proceedings ICASSP,* 1986.

[11] Woodland, P.C. and Povey, D., "Large Scale Discriminative Training of Hidden Markov Models for Speech Recognition," *Computer Speech and Language,* Vol. 16, pp.25-47.

[12] Malah, D., "System and Method for Noise Threshold Adaptation for Voice Activity Detection in Nonstationary Noise Environments," US Patent 5,991,718, Nov. 23, 1999.