CONTINUOUS ASR FOR FLEXIBLE INCREMENTAL DIALOGUE

C. Breslin, M. Gašić, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis and S. Young

cb404@cam.ac.uk
Cambridge University Engineering Department
Cambridge, CB1 2PZ, UK

ABSTRACT

Spoken dialogue systems provide a convenient way for users to interact with a machine using only speech. However, they often rely on a rigid turn taking regime in which a voice activity detection (VAD) module is used to determine when the user is speaking and decide when is an appropriate time for the system to respond. This paper investigates replacing the VAD and discrete utterance recogniser of a conventional turn-taking system with a continuously operating recogniser that is always listening, and using the recogniser 1-best path to guide turn taking. In this way, a flexible framework for incremental dialogue management is possible. Experimental results show that it is possible to remove the VAD component and successfully use the recogniser best path to identify user speech, with more robustness to noise, potentially smaller latency times, and a reduction in overall recognition error rate compared to using the conventional approach.

Index Terms— Dialogue system, ASR, VAD, POMDP, incremental ASR

1. INTRODUCTION

Spoken dialogue systems provide a convenient way for users to interact with a machine using speech. They are often deployed in noisy environments such as in-car applications, where the user does not have their hands free to interact in any other way.

Conventional dialogue systems typically use the idea of a dialogue turn being a user turn followed by a system turn. Only when the user has finished speaking does the system process the user input and take any action. This gives rise to a rigid model of turn-taking, which can be unnatural to users. There are many conditions under which users employ a more flexible turn-taking model, for example when they are under cognitive load and use more fillers, hesitations and barge-ins [1]. Furthermore, rigid turn-taking models often rely on a voice activity detection (VAD) component to decide whether the user is speaking or not, and this component can perform poorly, especially in noisy conditions, leading to confusion if the speech/non-speech classification is incorrect. To improve user satisfaction, several commercially deployed systems avoid the VAD problem by using a push-to-talk button, e.g. [2].

Recent research has suggested the use of an incremental dialogue system that allows for a more flexible turn taking model [3, 4, 5]. This paper investigates the replacement of the separate VAD and automatic speech recognition (ASR) components of a POMDP dialogue system [6] by a single continuously operating ASR component that is always listening in order to facilitate incremental dialogue. Experimental results show that it is possible to remove the VAD module entirely and, with appropriate training data, achieve an improvement in the detection of speech with potentially lower latency times. This framework gives the further advantage of increased flexibility and fewer components to adapt.

The outline of this paper is as follows. Section 2 discusses previous work in the area, section 3 discusses the incremental dialogue management using only the ASR output, section 4 presents experimental results, and section 5 draws conclusions.

2. EXISTING WORK

Incremental dialogue management has been proposed as a way of achieving a flexible turn-taking model in dialogue systems. This section discusses existing work on incremental dialogue management, and existing work on VAD.

2.1. Incremental dialogue

Recent work in dialogue systems has moved away from using a rigid model of turn-taking and towards using incremental ASR results to determine the best system action before the user has finished speaking [3, 5, 4]. Partial, or incremental, ASR results are periodically obtained while the user is speaking, and the dialogue manager decides whether to act on these partial hypotheses or not, based on all available information. In this paper, the term "dialogue manager" refers to everything downstream from ASR, to simplify the discussion. In reality, the dialogue manager of a POMDP system consists of several components such as semantic decoding, belief state update, TTS etc. An incremental dialogue system can enable modelling of conversational effects that are difficult to model with a strict turn-taking model, such as split utterances, users changing their mind, self-correction and hesitations, barge-in and backchannels on the part of both the system and the user.

Incremental ASR results have the problem that partial hypotheses are often unstable, particularly at the beginning of a word. That is, the words that are on the best path at one point in time may change by the next point in time. Thus, the best system action may also be unstable. The stability and accuracy of partial hypotheses has been measured using features collected from the decoding lattice [7, 8]. Such metrics can be used by the dialogue manager to decide whether to make use of a partial hypothesis or whether to discard it and wait until the next partial hypothesis is seen. The expected stability of partial hypotheses can also be improved. In [7], the points at which partial hypotheses are computed are carefully selected to be at times when the ASR either has high confidence in the current word or the language model end of utterance symbol has been reached. In [9], additional right context is included before a partial hypothesis is returned, which introduces a short lag but improves stability.

One final decision to be made in an incremental dialogue system is when the system should respond to the user. The system can respond immediately as in [5], thus potentially interrupting the user, or wait until the end of a user utterance. In [10], prosodic and syntactic features were used to predict the end of a user utterance without having to wait for a user silence. Alternatively, the system can estimate the optimal moment to barge in based on, for example, a measure

of information density in the user's utterance [11]. Such prediction is useful in allowing incremental dialogue systems to respond at appropriate times with minimal delay.

Despite the issues with using partial ASR hypotheses, incremental dialogue systems have been shown to be favoured by users, even if they do not yield improved objective performance, due to their speed and naturalness [3]. However, the system architecture is complicated by the use of an incremental dialogue manager which receives potentially conflicting information about macro turns from the VAD module, and about micro turns from the ASR module.

2.2. Voice Activity Detection

Voice activity detection is typically a computationally efficient preprocessing step to classify audio frames as either speech or non-speech, so non-speech frames can be discarded and only speech segments are passed downstream to the speech recogniser. The most popular approach is to use speech and non-speech Gaussian mixture models (GMMs) trained on appropriate data [12, 13], and to assign each audio frame to the class which has the highest likelihood. Other classifiers such as SVMs [14] and MLPs [15] have also been used. This classification approach has been used successfully in LVCSR tasks such as meetings [16, 15, 17] and broadcast news [18]. To improve on the use of a single classifier, a hybrid approach was proposed that uses a threshold on the energy of the audio to first discard very low energy frames [19]. Then, only the high energy frames are passed to the GMM classifier. The threshold is relative to the silence level in the audio, which is estimated as the audio progresses.

However, VAD is often one of the weakest components of an ASR system, particularly in noisy environments that are mismatched to the training data. In ASR tasks, it is impossible to recover from VAD errors where speech is incorrectly classified as non-speech, but non-speech frames classified as speech are passed to the recogniser and so can be transcribed as silence at the recognition stage. Hence, VAD modules are often tuned to give a high recall of speech, at the price of low precision. In a dialogue system where the VAD result is also used as an input to the dialogue manager to control turn-taking, incorrectly identifying noise as speech can lead to confusion where the system thinks the user is speaking but the user is, in fact, silent.

Use of a VAD component is an easy way of reducing the computation done by the speech recogniser when computational resources are limited. As available computational power increases, however, and the move is made towards server based systems, there is less need to reduce the computational load of the ASR module. It has long been acknowledged that the speech recognition component itself is a far better speech/silence detector than a simple speech/silence classifier since it has a much more accurate model of speech. Recent efforts have been made to use feedback from the speech recognition to adapt and improve VAD performance. In [20], input features to the VAD classifier were derived from the state output distributions of an LVCSR HMM set. Full decoding was not performed, but instead a subset of likely Gaussians from the HMM output distributions were used to compute approximate likelihoods for broad phone classes, and these were fused with energy based features for use in a GMM classifier. In [21], high confidence speech and silence segments identified by the recogniser were used to adapt the VAD models online, yielding gains in ASR performance. These methods make use of acoustic models to improve the performance of the VAD models.

This paper lays the framework for a flexible incremental dialogue system that has no explicit VAD module. In contrast to previous work, the ASR recogniser is listening continuously throughout the dialogue without the use of a separate VAD classification stage. Such an architecture simplifies the incremental dialogue manager which no longer has to interface with two, potentially conflicting, modules, allows for more flexible turn-taking models to be developed, and allows easy adaptation to noisy environments using powerful techniques developed for noisy ASR. This idea is discussed in more detail in the following section.

3. INCREMENTAL ASR WITHOUT VAD

There are many advantages to removing the VAD component of a dialogue system and relying solely on the ASR output. Reducing the number of system components leads to a simpler architecture that is easier to implement and test. Furthermore, using an always-on-ASR component provides a framework in a dialogue system for a more flexible turn-taking model. Dialogue turn-taking is complex, and requires the system to recognise and make use of many verbal and non-verbal cues. Recognition features can easily be included when the system is making a decision about when to speak. The dialogue manager can inspect partial ASR hypotheses or partial lattices at appropriate times, before deciding whether to take action. Other cues, such as prosody, can be integrated into the decision making process to better determine the end of a user turn.

Performance of both ASR and VAD components are often degraded in noisy conditions, and so a key advantage of combining VAD and ASR is the ability to adapt both at the same time, instead of separately. Hence, more advanced speaker and noise robustness techniques such as CMLLR, MLLR [22], VTS [23] or PCMLLR [24] can be used to directly improve VAD performance alongside ASR.

When removing the VAD component, the ASR best path can be used to guide the dialogue system. Figure 1 shows the user of a system uttering the phrase *"Chinese <pause> in the centre"*. The recogniser best token at each frame (speech/silence) is shown below. At point (a) the user has begun to speak and speech is detected on the recogniser best path. There is a short lag between the user starting and ending speech before the corresponding best token reflects the true state, but the best token at this point can be traced back to find the actual beginning of the speech. At point (b) silence is detected as the user has paused in the middle of the utterance, and at point (c), speech is again detected on the best path as the user has resumed speaking. At point (d), there is silence again as the user has finished speaking, and after a small number of frames, e.g. 30, there is still silence on the best path and so the system can be confident that the user has finished speaking. This is point (e) on the graph.

The dialogue system can then behave differently depending on the point in a user turn. For example, points (b) and (d) can be used to pass the partial hypothesis to the dialogue manager to begin preparing a response or to utter a backchannel. The belief propagation algorithm for updating estimates of the dialogue state is a loopy iterative algorithm. At points (b) and (d), the current partial utterance can be used to update the belief state, thus altering the dialogue belief state before the user has finished speaking and before the system has decided to take action.

To avoid interrupting the user, the system may delay responding until point (e). This introduces a lag between the end of user's speech and the system response, but this can be mitigated by performing processing at points (b) and (d) and waiting until point (e) is reached before responding. Points (b) and (d) can also be appropriate places to utter backchannels or to interrupt the user. If the system has determined that there is no match to the user's request, e.g. no Chinese restaurant that satisfies earlier constraints, the system can interrupt the user to tell them so without waiting for the end of the user utterance.

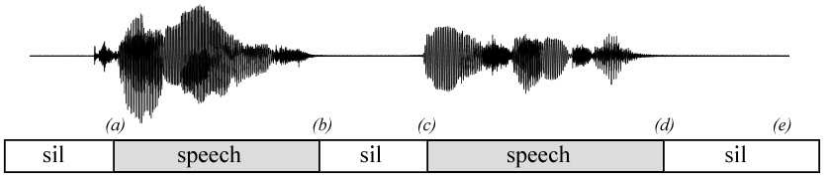

Fig. 1. System user speaking the utterance *"Chinese <pause> in the centre"'* and recogniser best path

Such strategies do not make full use of the ASR decoding result. It is often possible to predict the word being spoken before the user has finished speaking it, as the language model encodes some right-context. For example, the user may be part-way through speaking the word *"Chinese"* and the recogniser is confident that this is the word being spoken. In this case, as soon as the recogniser is confident of the current word, the partial ASR hypothesis can be passed downstream to the dialogue manager before the user has finished speaking.

The use of confusion networks rather than 1-best or N-best word strings has been shown to improve semantic decoding accuracy in conventional turn taking systems [25]. The same performance gains can be obtained in an incremental system since at any point in time t a partial ASR decoding result can be returned in the form of a word lattice or confusion network spanning the previously detected start of the user utterance upto t.

This section has explained some of the ways in which the ASR decoding result can be used to improve dialogue system naturalness and responsiveness. The following section presents experimental results to demonstrate that an always-listening ASR module performs better than a VAD module for detecting user speech, especially in noisy data, with only a small change in ASR performance, thus enabling these techniques to be used in dialogue systems.

4. EXPERIMENTAL RESULTS

Experiments were carried out using audio data collected using a restaurant information dialogue system deployed over a phone line. This is a medium vocabulary ASR task where the ASR output feeds into a statistical POMDP dialogue system. Two live trials were performed to collect data sets `GM1` and `GM2`. The first was collected in a stationary car and the second in a moving car, each set was split into a) a dev set used in training and b) a test set. Table 1 shows the amount of data in each set. A large portion of the audio is silence where the user is listening to the system speak.

	GM1a	GM1b	GM2a	GM2b
Audio (hrs)	8.5	5.8	9.7	5.8
Speech (hrs)	1.7	0.7	1.3	0.8
# Dialogues	361	242	467	312

Table 1. Data sets

Acoustic models were trained using 76 hours of narrowband conversational speech, where speech segments with a small amount of pre- and post-silence were extracted from the audio. Next, MAP adaptation was used to adapt the models to task specific data, collected during trials, followed by MPE training with the combined data set. There are an average of 8 GMM components per HMM state, with the number of components per state being proportional to the amount of training data for that state. Each of the silence states had roughly 30 components.

A language model was trained on 410k words of transcribed speech from previous trials. A general background model was not found to be helpful as the things that people tend to say in the context of the dialogue system are very limited, and well covered by data from previous trials. Thus the perplexity of the language model is very low, around 6-14.

In order to perform ASR over whole dialogues, the language model needs to allow for the silences between utterances. For decoding segments and whole dialogues, different language models were built. The first uses one entry per utterance in the dialogue with a start and end of utterance symbol `<s>`, `</s>`, while the second uses one entry per dialogue, with separate markers for the start and end of utterance `<u>`, `</u>`, and start/end of dialogue `<s>`, `</s>`.

Segment language model:

```
<s> hello i'd like an italian restaurant
please </s>
<s> cheap </s>
<s> where is it </s>
<s> ok thank you good bye </s>
```

Whole-dialogue language model:

```
<s> <u> hello i'd like an italian
restaurant please </u> <u> cheap </u> <u>
where is it </u> <u> ok thank you good
bye </u> </s>
```

Two configurations were compared. First, a conventional configuration was tested consisting of a VAD component feeding audio segments to an ASR component. In the second test, ASR was performed on the whole dialogue with no separate VAD component. A key advantage of using ASR only is being able to adapt both the VAD and the ASR in one step. Thus the goal was to perform offline adaptation and adapt the system to the noisy data in set `GM2`. Baseline VAD GMM models were trained on whole dialogues collected during trials, including all the silence between user utterances. The VAD GMMs had 256 components in the silence state and 128 in the speech state. Initial VAD models (V1) were trained on clean data, including the `GM1a` set. Noisy VAD models,V2, were trained by also using the noisy `GM2a` data.

Table 2 shows that the inital VAD models, V1, perform well on the matched clean dataset `GM1`, but their accuracy degrades on the noisy `GM2` data. The noisy VAD models, V2, have improved performance on the noisy set `GM2`. For example, frame correctness for the noisy test set `GM2b` rises from 84.3% to 89.8% when moving from VAD models V1 to V2. Much of the error in the VAD accuracy arises as a consequence of noise being erroneously classified as speech. On set `GM2b` when the frame correctness is 89.8%, speech recall is 94.5% and speech precision is 61.4%.

ASR models A1 are trained on all data, and both the clean `GM1a` and noisy `GM2a` dev sets were used as task specific data to perform the final MAP adaptation and MPE stages. Table 2 also shows the VAD results when these acoustic models were used to decode the whole dialogue audio, and the recogniser best path was used to identify speech segments as described in section 3. The A1 ASR models perform worse than the standalone VAD models, with a frame cor-

VAD	ASR	GM1a dev	GM1b test	GM2a dev	GM2b test
V1		93.7	93.5	82.9	84.3
V2		95.3	95.5	89.5	89.8
-	A1	75.9	73.0	72.5	76.0
-	A2	98.0	98.3	95.5	96.6

Table 2. VAD performance, Frames Correct (%)

rectness of 76.0% on the noisy test set GM2b, and tend to identify many non-speech segments as speech.

To address this poor performance, the decoding hypotheses using A1 models were used to identify segments of the audio that led to erroneous insertions in the dev sets GM1a and GM2a. That is, those ASR insertions that are hesitations or probable noise, but are not immediately adjacent to actual segments of speech. A total of 5 hours of non-speech segments were collected this way, which were then used during the MAP and MPE stages to train improved acoustic models A2. The silence models for A2 were not updated during the MPE training stage since updating them was found to degrade performance. Table 2 shows that these new acoustic models lead to large improvements in VAD performance, and outperform the standalone VAD models, particularly on the noisy data where the frame classification rate on the noisy test set was 96.6%.

These results show that a simple GMM classifier for identifying speech and non-speech in noisy data becomes unreliable in noise, even when the GMMs are trained on noisy data, and that appropriately trained acoustic models can give a more reliable indicator of user speech.

Table 3 shows the ASR word error rates for both scenarios. As expected, results for the clean GM1 set are better than the noisy GM2 set, by roughly 10% absolute. ASR results are scored over the whole dialogue, so a large source of errors results from inserted segments where the audio contains non-speech. That is, where noises and non-verbal sounds from the user are erroneously transcribed as speech.

VAD	ASR	GM1a dev	GM1b test	GM2a dev	GM2b test
V1	A1	23.3	29.7	29.1	38.1
V2	A1	23.1	29.4	27.0	37.5
V2	A2	19.7	24.4	22.4	34.0
-	A1	28.4	34.5	41.0	44.3
-	A2	19.3	23.8	22.4	32.9

Table 3. ASR performance, Word Error Rate (%)

When using VAD+ASR moving from the V1 to V2 VAD models, there is a small gain in ASR performance, 38.1 to 37.5% WER on the GM2b test set. Then, when using the improved A2 acoustic models, the improved performance of 34.0% is achieved on that set.

As expected because the A1 acoustic models had poor VAD performance, they also have poor ASR performance when used continuously with no VAD module. This is due to the large number of insertions from noises and other non-vocal sounds. Finally, ASR results show that using just acoustic models A2, the performance achieved is comparable to that when using the standalone VAD models followed by ASR on each segment. For example, on the noisy test set GM2b, performance improves from 34.0% to 32.9% when moving from VAD+ASR to ASR only. This small improvement is unlikely to be noticeable to the user of a statistical dialogue system, and a greater effect on user satisfaction is likely to come from the improved VAD performance leading to better turn-taking.

These results show that offline adaptation of ASR models to noisy data and using continuous ASR works better than adapting the ASR and the VAD models separately. The ASR models are more robust to noise and the ASR best path can be successfully used for voice activity detection, allowing for system behaviour such as that described in the previous section.

	VAD+ASR	ASR
Reference segments identified	86%	83%
Inserted segments	63%	39%
Average error detecting end of speech	108ms	45ms
Expected latency	>608ms	<345ms

Table 4. Analysis of detected speech segments using V2 and A2 models on noisy test set GM2b

Table 4 shows an analysis of the segments predicted by both approaches, using the V2 and A2 models on the noisy test set GM2b. The VAD models correctly identify more of the reference segments than the ASR models alone, 83% compared to 86%, but the VAD models also insert far more erroneous segments. 63% of the segments hypothesised by the VAD models are non-speech segments, compared to only 39% of the segments predicted by the continuously listening ASR models. The average error in detecting the end of speech is also shown, for the correctly detected speech segments, and is larger for the VAD models than for using the ASR decoding path. Finally, the average error in the end border can be used to predict the expected latency in responding to the user. Using separate VAD models, a lookahead window of 500ms has been found to give good performance and this leads to an average delay of 608ms after the user has finished speaking before the system can start preparing a response. In contrast, the expected latency when using the ASR path is shorter. In this paper, a conservative delay of 300ms has been used to identify a segment as speech, i.e. the time between points (d) and (e) in figure 1. However, all processing can be done at point (d), leaving the system ready to respond as soon as the end of a user utterance has been confirmed. This 300ms is a conservative figure, and in practice the system can be ready to respond much earlier, and even before the user has finished speaking if the final word is well predicted by the language model.

5. CONCLUSIONS

Conventional dialogue systems normally use a cascade of VAD and ASR components to identify when the user is speaking and what they are saying. This imposes an artificial rigid turn-taking model which is unnatural to users and can lead to poor user satisfaction when using a spoken dialogue system.

This paper has investigated replacing the VAD+ASR cascade with a single ASR component that is always listening, and using the real-time decoded output to identify user speech. This more flexible framework allows for more robust adaptation to noisy environments and for finer control over turn-taking during dialogues.

Experimental results showed an improvement in VAD performance when using acoustic models adapted for the noisy conditions, a small improvement in ASR performance, and potential for much shorter system response times. As statistical dialogue systems have an inbuilt resiliance to ASR errors, it requires a large change in ASR performance (perhaps 10% absolute) for the user to notice any effect. Thus the improved speech activity detection performance and responsiveness are expected to improve user satisfaction.

Future work will involve moving this work from offline recognition and adaptation to an online dialogue scenario and combining it with an incremental turn-taking model.

6. REFERENCES

[1] M. Gašić, P. Tsiakoulis, M. Henderson, B. Thomson, K. Yu, E. Tzirkel, and S. Young, "The Effect of Cognitive Load on a Statistical Dialogue System," in *SIGDIAL*, 2012.

[2] H. Cheng, H. Bratt, R. Misra, E. Shriberg, S. Upson, J. Chen, F. Weng, S. Peters, L. Cavedon, and J. Niekrasz, "A Wizard of Oz Framework for Collecting Spoken Human-Compute Dialogs," in *Interspeech*, 2004.

[3] G. Aist, J. Allen, E. Campana, C. Gomez-Gallo, S. Stones, M. Swift, and M. Tanenhaus, "Incremental understanding in human-computer dialogue and experimental evidence for advantages over non-incremental methods," in *Proc. Workshop on the Semantics and Pragmatics of Dialogue (DECALOG)*, 2007.

[4] D. Schlangen and G. Skantze, "A General, Abstract model of incremental dialogue processing," *Dialogue and Discourse*, vol. 2, pp. 113–141, 2012.

[5] E. Selfridge, P.A. Heeman, I. Arizmendi, and J.D. Williams, "Demonstrating the incremental interaction manager in an end-to-end "Let's Go!" dialogue system," in *SLT*, 2012.

[6] B. Thomson and S. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Computer Speech and Language*, vol. 24, pp. 562–588, 2010.

[7] E. Selfridge, I. Arizmendi, P. Heeman, and J.D. Williams, "Stability and Accuracy in Incremental Speech Recognition," in *SIGDIAL*, 2011.

[8] I. McGraw and A. Gruenstein, "Estimating word-stability during incremental speech recognition," in *Interspeech*, 2012.

[9] T. Baumann, M. Atterer, and D. Schlangen, "Assessing and improving the performance of speech recognition for incremental systems," in *NAACL-HLT*, 2009.

[10] M. Atterer, T. Baumann, and D. Schlangen, "Towards incremental end-of-utterance detection in dialogue systems," in *22nd International Conference on Computational Linguistics*, 2008.

[11] N. Dethlefs, H. Hastie, V. Rieser, and O. Lemon, "Optimising Incremental Dialogue Decisions using Information Density for Interactive Systems," in *Proceedings of the Conference on Empirical Methods in Natural Language processing*, 2012.

[12] S. Tranter and D. Reynolds, "An overview of automatic speaker diarisation systems," *IEEE Trans on Audio, Speech and Language Processing*, vol. 14, pp. 1557–1565, 2006.

[13] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker Diarization: A Review of Recent Research," *IEEE Trans on Audio, Speech and Language Processing*, vol. 20, pp. 356–370, 2012.

[14] A. Temko, D. Macho, and C. Nadeu, "Enhanced SVM training for robust speech activity detection," in *ICASSP*, 2007.

[15] T. Hain, L. Burget, J. Dines, P. Garner, A. el Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "The AMIDA 2009 Meeting Transcription System," in *Interspeech*, 2010.

[16] H. Sun, B. Ma, S.Z.K. Khine, and H. Li, "Speaker Diarization System for RT07 and RT09 Meeting room audio," in *ICASSP*, 20010.

[17] C. Breslin, K.K. Chin, M.J.F. Gales, and K. Knill, "Integrated Online Speaker Clustering and Adaptation," in *Interspeech*, 2011.

[18] J.L. Gauvain, L. Lamel, and G. Adda, "Partitioning and Transcription of Broadcast News data," in *ICSLP*, 1998.

[19] X. Anguera, C. Wooters, M. Anguilo, and C. Nadeu, "Hybrid speech/non-speech detector applied to speaker diarization of meetings," in *Proc Speaker Odyssey Workshop, Puerto Rico*, 2006.

[20] E. Marcheret, K. Visweswariah, and G. Potamianos, "Speech Activity Detection fusing acoustic phonetic and energy features," in *Interspeech*, 2005.

[21] K. Thambiratnam, W. Zhu, and F. Seide, "Voice activity detection using speech recognizer feedback," in *Interspeech*, 2012.

[22] M.J.F. Gales and P.C. Woodland, "Mean and Variance Adaptation within the MLLR Framework," *Computer Speech and Language*, vol. 10, pp. 249–264, 1996.

[23] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM Adaptation using vector Taylor series for noisy speech recognition," in *ICSLP*, 2000.

[24] R. van Dalen and M.J.F. Gales, "Predictive linear transforms for noise robust speech recognition," in *ASRU*, 2007.

[25] M. Henderson, M. Gašić, B. Thomson, P. Tsiakoulis, K. Yu, and S. Young, "Discriminative Spoken Language Understanding using Word Confusion Networks," in *SLT*, 2012.