THE 2013 BBN VIETNAMESE TELEPHONE SPEECH KEYWORD SPOTTING SYSTEM

Stavros Tsakalidis, Roger Hsiao, Damianos Karakos, Tim Ng, Shivesh Ranjan, Guruprasad Saikumar, Le Zhang, Long Nguyen, Richard Schwartz, John Makhoul

Raytheon BBN Technologies, Cambridge, MA, USA

ABSTRACT

In this paper we describe the Vietnamese conversational telephone speech keyword spotting system under the IARPA Babel program for the 2013 evaluation conducted by NIST. The system contains several, recently developed, novel methods that significantly improve speech-to-text and keyword spotting performance such as stacked bottleneck neural network features, white listing, score normalization, and improvements on semi-supervised training methods. These methods resulted in the highest performance for the official IARPA Babel surprise language evaluation of 2013.

Index Terms— stacked bottleneck neural network features, keyword spotting, white listing, score normalization

1. INTRODUCTION

Keyword spotting (KWS) is the task of detecting the occurrences of words in a speech signal. State-of-the-art KWS systems couple speech-to-text (STT) technology with traditional text-matching techniques [1]. A detailed survey of existing KWS techniques can be found in [2, 3].

In this paper, we present a keyword spotting system for conversational telephone speech that BBN constructed in response to the NIST Babel Surprise Language evaluation of 2013. The system contains several, recently developed, novel methods that significantly improve STT and KWS performance. Acoustic feature extraction is based on a Stacked Bottleneck (SBN) neural network (NN) architecture [4] that significantly outperforms PLP and Bottleneck (BN) features. Improvements on semi-supervised training via confidence weighted training, semi-supervised discriminative training and semi-supervised multilayer perceptron (MLP) training are described that help semi-supervised training for low resource languages and high WER environments [5].

In addition to improving fundamental automatic speech recognition (ASR), we also advanced the core KWS technology, which goes well beyond ASR technology. The novel methods used in the KWS component of the system are designed to improve keyword recall and keyword score accuracy. Typically, in order to increase keyword recall, we look for hits for the keywords in the recognition lattice, which provides many alternatives. We can increase the beam width in the search and increase the depth of the lattice, but this can quickly increase the computation and memory to an unacceptable point. An alternative solution to this problem is to inform the recognizer of the set of keywords and protect those keywords from being pruned out so that they almost always be in the lattice if they were actually spoken. We call this list of keywords a white list [6] in that these words are (almost) always accepted, albeit with a low score. In this paper, we introduce a new extension to white listing that can be applied in situations where the keywords are not known prior to decoding.

Another challenge we addressed was to produce a score for each hit that is consistent across all keywords. Score normalization, based on a machine-learning framework [7], produces scores that are commensurate across keywords. This significantly improves KWS accuracy and benefits many applications of KWS such as Boolean queries and word clouds for summarization.

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. 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.

2. SPEECH RECOGNITION COMPONENT

2.1. Acoustic Features

Neural networks were used to generate BN or SBN features [8]. A detailed description is available in [4]. The SBN structure contains two NNs: the BN outputs from the first one are stacked, downsampled, and taken as an input vector for the second NN. This second NN has again a BN layer, of which the outputs are taken as input features for the recognition system. The input features of the first NN are 15 critical-band energies obtained with a Mel filter-bank, with conversation-side-based mean subtraction applied. 11 frames of these features are stacked and a Hamming window multiplies the time evolution of each parameter. Finally, DCT is applied, of which 0th to 5th coefficients are retained, making the size of the feature vector $15 \times 6 = 90$.

The sizes of the both NNs were set to 1M weights for most of the experiments. When the best input features, structure and normalization were found, NN sizes were increased to 2M weights. Both NNs were trained to classify phoneme states (3 states per phoneme). These targets were generated by forced alignment with baseline PLP models and stayed fixed during the training. The final feature stream was built by concatenation of PLP-HLDA (39 dimensions), SBN (30) and pitch along with the first and second derivatives (3) adding up to final dimensionality of 72. Then, region dependent transformation (RDT) [9] is performed to estimate a discriminative feature projection to reduce the dimension to 46.

2.2. Acoustic and Language Modeling

The ASR system uses BBN's Byblos speech recognizer [10] which models speech as the output of context-dependent phonetic Hidden Markov Models (HMMs). The outputs of the HMM states are mixtures of multi-dimensional diagonal Gaussians. Different forms of parameter tying are used in Byblos, including State Tied Mixture (STM) triphone and State Clustered Tied Mixture (SCTM) quinphone models. The mixture weights in both these cases are shared

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

based on decision tree clustering using linguistic rules using phonetic questions.

Decoding is carried out in a multi-pass search strategy. The forward pass uses a STM model and a bigram language model, and outputs the most likely wordends at each frame together with their scores. The backward pass then uses the output of the forward pass to guide a Viterbi beam search with a SCTM within-word quinphone acoustic model and a trigram language model. A lattice is also generated. Finally, we do lattice rescoring using a SCTM cross-word quinphone model. The top scoring hypothesis represents the system's recognition output. The acoustic model is a speaker adaptive model [11] trained discriminatively under the boosted maximum mutual information (BMMI) criterion [12]. The language model (LM) is a word-based trigram LM with modified Kneser-Ney smoothing.

3. SEMI-SUPERVISED TRAINING

One of the main goals of the IARPA Babel program is rapid development of speech technologies for new languages with limited resources. Under the Limited training condition, the program provides 10 hours of transcribed audio and under 100 hours of untranscribed audio. Semi-supervised training provides a mechanism for improving system performance using unsupervised (untranscribed) data given a relatively small amount of supervised (transcribed) data. The basic approach for semi-supervised training is first building a bootstrap model using some supervised data, and then using this model to transcribe the unsupervised data. This automatically transcribed data is then used to supplement the supervised data for building the final model.

The initial supervised systems under the limited transcribed data condition often have over 70% word error rate (WER). Since the performance of the bootstrap system is particularly poor, the automatic transcription may contain mostly errors. In addition, using transcriptions with high error rates may have more impact on discriminative training, which tries to minimize the errors against the reference transcriptions. The difficulties of semi-supervised discriminative training have been discussed in [13, 14, 15].

Therefore, we revisited semi-supervised training and introduced three techniques to address this acute condition: (i) a confidence weighted training method which uses a confidence model to select data and also weighs the supervised and unsupervised data, (ii) a semi-supervised discriminative training technique that handles the errors in the automatic transcriptions, and (iii) a semi-supervised MLP training for acoustic feature extraction. A detailed description of the semi-supervised techniques is available in [16].

3.1. Confidence Weighted Training

Figure 1 is an overview of the confidence weighted training. This training procedure consists of two parts: (1) unsupervised data selection followed by (2) semi-supervised acoustic model training.

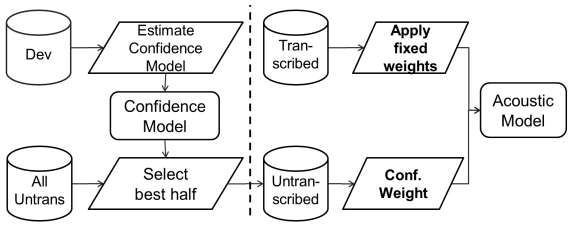

Fig. 1. Overview of the confidence weighted training.

The data selection procedure for semi-supervised training is described in [5]. First, the untranscribed audio data is segmented into utterances using a speech activity detection system which is trained on the 10-hour training corpus using an architecture similar to [17]. It is then decoded using the system trained on the same 10-hour manually transcribed corpus. The confidence of each utterance is computed based on a confidence model trained on the development set. Finally, the best half of the utterances are selected according to their confidence scores for acoustic model training.

The utterance based confidence scores are then converted into weights based on $w_i = s \times c_i + b$ where w_i is the weight for utterance i, s the slope, c_i confidence score for utterance i, b the bias. In this work, s is 2.0 and the average of the utterance-level weights is constrained to one. Hence, $b = 1 - \frac{\sum_{i=1}^{N} s \times c_i}{N}$ with N being the total number of utterances. The posterior probabilities used in training are then multiplied by these weights, so utterances with lower confidence would contribute less to the collective statistics.

3.2. Semi-supervised Discriminative Training

Our proposed semi-supervised discriminative training aims to focus on the supervised data, for which errors can be accurately located. While the small amount of supervised data may not allow us to estimate the model parameters reliably, we use the unsupervised data as a constraint to control the optimization. Figure 2 is an overview of our semi-supervised discriminative training.

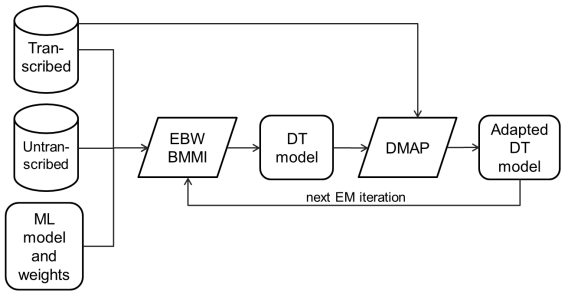

Fig. 2. Overview of semi-supervised discriminative training.

The idea is to enforce the output model to be close to the estimate using the entire data set, which is likely to be more robust. This is achieved by first estimating the model using the entire semi-supervised set. Then, we adapt the model discriminatively using the supervised data. Adaptation is performed via discriminative MAP adaptation (DMAP) [18].

3.3. Semi-supervised Multi-layer Perceptron Training

The semi-supervised MLP training method consists of two stages: The first stage is to train the MLP using only the supervised data. This MLP is then used to train an ASR system to filter and select unsupervised data. The selected set is then used to train the final MLP. We found that it was best to use all of the data rather than discard the data with low confidences. Details of the semi-supervised MLP training are available in [4, 19].

4. KEYWORD SEARCH COMPONENT

4.1. White Listing

A good ASR system is essential but not sufficient for building KWS systems. An ideal KWS system should have high recall for every

possible keyword. We recently introduced a simple but effective solution to this problem, which is to inform the recognizer of the set of keywords and protect those keywords from being pruned out so that they will almost always be in the lattice if they were actually spoken. We call this list of keywords a white list [6] in that these words are (almost) always accepted, although possibly with a low score.

In order to avoid pruning errors, we implemented a white list feature in the recognizer. During normal beam pruning, we compare the score at a state with the score of the best hypothesis at that instant and remove the state from consideration if its score is below some threshold relative to that highest score (the beam). In this case, if the state belongs to one of the words in the white list, the threshold is much lower (a wider beam) so that it becomes very unlikely for this keyword to be pruned out when it is in fact in the speech. This modification is made to all of the places in the decoder where any pruning takes place. Even though the computation for the keywords is increased by a large factor, the overall computation does not increase by much, since the number of keywords is typically much less than the number of words in the decoding lexicon.

In situations where the keywords are not known in advance we typically run recognition using only the provided dictionaries. Since we do not know the keywords, we cannot white list them. However, in order to make sure the system will be able to find the keywords later, we want to make sure we have a substantial number of hits for most of the lexical items. As a natural extension, the white listing can be generalized to apply to any lexical term. Adding all the lexical terms in the white list is equivalent to increasing the beam and the depth of the lattice which is impractical. Therefore, the system should be able to automatically select a subset of the lexical terms to be included in the white list so that we have a substantial number of hits for most of the lexical items.

In this phase, the lexical term selection was based on the following algorithm: We first run the KWS system on a transcribed held-out set and perform keyword search for every lexical term. Then, based on the KWS results, we add to the white list any lexical terms that (a) appear in the audio and have low recall and low hits or (b) do not appear in the audio and do not have a substantial number of hits. The condition for adding a lexical term in the white list can be expressed by the following expression

$$\text{if}(((N_{true} \geq 1)\text{AND}(\#\text{hits} < H_1)\text{AND}(\text{recall} < R)) \\ \text{OR}(\#\text{hits} < H_2)) \quad (1)$$

where N_{true} is the number of reference occurrences of the word. The thresholds R, H_1, H_2 are determined manually, for now, so that the resulting white list size is kept reasonably low.

4.2. Keyword Search

The speech recognition system produces a detailed lattice of word hypotheses. The resulting lattice is annotated with acoustic, language and pronunciation model scores. Then a forward-backward pass is applied on the lattice to estimate the posterior probability for each word arc in the lattice. Word lattices are further expanded to sub-word units, such as characters and phones, by splitting each word into its sub-units. This allows for searching out-of-vocabulary (OOV) words and words that were missed by the whole-word search. Then, the lattices are converted to consensus networks (c-net) which provide a simplified method for finding keywords as sequence of phones or words. C-nets require orders of magnitude less storage for indexing compared to lattices.

Recall can be improved significantly by allowing approximate matches within the c-net to be returned by allowing substitutions, deletions, and insertions with appropriate confusion penalty. The target query represented by a sequence of units (phones, characters, etc.) are aligned to the c-nets using dynamic programming [20]. A confusion matrix, estimated on phone sequences from a c-net generated over training data, is used for keyword search. The score of the keyword is taken as either the product or the geometric mean of the scores of the individual posteriors in the c-net.

4.3. Score Normalization

Score normalization is the process through which the original scores of the detections of different keywords are made commensurate with each other. This is necessary in order to create global ranked lists of hits, across all keywords, so that those hits which are ranked higher have a higher probability of being correct. There are many applications of KWS that require comparable scores across keywords such as Boolean queries and word clouds for summarization. Even when we strive to get good estimates of keyword confidence, the confidence scores for different keywords tend to vary systematically for several reasons. Thus, to obtain state-of-the-art KWS performance, it is essential to use score normalization.

Our most advanced score normalization method is based on a machine-learning framework that utilizes many features. A detailed description is available in [7]. The original scores of the detections go through a number of transformations, such as: rank-normalization, mapping-back to posteriors, "probability of correct" normalization $p_{corr}()$ and through some non-linear functions such as $log()$, $()^{1/2}$, $()^2$, sigmoid. The $p_{corr}()$ mapping is estimated by sorting all hits by score, defining bins, and then computing the probability that a random detection in the bin is correct. Then, the transformed scores, together with various additional features, e.g., keyword training count, keyword length, conversation-aggregated scores, are concatenated together into a feature vector. This vector, together with a target variable denoting whether the detection is a true positive or a false alarm, is given as input to Powells method [21], which learns a linear model.

5. EXPERIMENTAL RESULTS

5.1. Data and Evaluation Conditions Description

The Babel training data is mainly conversational speech between two persons on a telephone channel, but it also contains a small amount of read speech. The telephone channels can be landlines, different kinds of cellphones, or phones embedded in vehicles, and the sampling rate is 8000 Hz. In the first year of the program Vietnamese was chosen to be the language for open evaluation. This effort uses the IARPA Babel Program Vietnamese language collection release IARPA-babel107b-v0.7. The development set consists of roughly 10 hours of conversational telephone speech and the evaluation set contains around 75 hours of data. The evaluation keyword list consists of 4065 keywords. Each keyword may contain several words and it may or may not be in the training vocabulary.

The evaluation has several different conditions that depend on the amount and source of the training data sets and whether processing of the audio after knowledge of the keywords was used or not. In the full language pack (FullLP) condition, the training data consist of 100 hours of transcribed audio. In the limited language pack (LimitedLP) condition, the training data consist of a 10-hour subset of the transcriptions and lexicon of the full language pack and all the audio data of the full language pack. The lexicon was only from the limited language pack and contained 3117 words. Another differentiator depends on the knowledge of the keywords prior to decoding.

In the no test audio re-use (NTAR) condition the system does not reprocess the test audio after keywords are provided. In the test audio re-use (TAR) condition the system re-processes the audio with knowledge of the search keywords.

Accuracy is judged relative to a time-marked reference transcript. A system detection is considered correct if a corresponding exact orthographic match of the term appears in the reference transcript within 0.5 seconds of the asserted time. System accuracy on a given collection of query terms is measured by the Actual Term-Weighted Value (ATWV) metric [22]

$$\text{ATWV} = 1 - \frac{1}{K}\sum_{w=1}^{K}\left(\frac{\#\text{miss}(w)}{\#\text{ref}(w)} + \beta\frac{\#\text{fa}(w)}{T - \#\text{ref}(w)}\right) \quad (2)$$

where K is the total number of keywords that occur in the test set, $\#\text{miss}(w)$ is the number of true tokens of keyword w that are not detected, $\#\text{fa}(w)$ is the number of false detections of w, $\#\text{ref}(w)$ is the number of reference tokens of w, T is the total number of trials (e.g., seconds in the audio), and β is a constant, set at 999.9.

5.2. MLP Features Results

An important contributing factor to our high performing ASR and KWS system is the use of MLP features. The MLP features were provided by our partners at Brno University of Technology (BUT). Table 1 compares the PLP to the MLP front end, as described in Section 2.1. Both systems are build according to the method described in Section 2.2. The MLP features significantly improve performance by 8.7% absolute in WER and 11.5% aboslute in ATWV.

Front End	WER (%)	ATWV
PLP	58.5	0.423
MLP	49.8	0.538

Table 1. PLP and MLP front end on the Dev set for FullLP NTAR.

5.3. White Listing Results

In this section we analyze the effectiveness of white listing, as described in Section 4.1. We run the KWS NTAR system on the Dev set and then searched for all lexical terms. Based on the keyword retrieval results we selected 1500 lexical terms for white listing according to the formula 1. For words that appear in the reference transcripts we set the target recall level R to 70% and the minimum #hits to 50 (H_1). For words that do not appear in the references, we require a minimum of 10 hits (H_2).

Table 2 compares the ATWV, recall and c-net density (arcs per second) statistics with and without white listing. The baseline system does not use a white list but has higher than usual beam widths for decoding and therefore generates big c-nets (122 arcs/sec). Even so, the recall rate of the baseline system does not exceed 74%. Moreover, there are 257 keywords with no hits. By white listing the 1500 lexical terms we increase the recall to 84% and decrease the number of keywords with no hits to 84. The c-net density increases by 30% and the ATWV performance increases by 2% points.

The last two rows of Table 2 compare the performance of the NTAR to the TAR condition by using white listing. Note that the TAR condition assumes the knowledge of the keywords prior to decoding. Therefore, for the TAR condition, we add all keywords to the white list. We observe that by using white listing for the NTAR condition we are able to obtain half of the total gain obtained by knowing the keywords prior to decoding. Therefore, the use of white listing in the NTAR condition enabled us to reduce the gap between the NTAR and TAR condition.

Configuration	Recall (%)	C-net density	#kwds w/o hits	ATWV
w/o wl (NTAR)	74	122	257	0.538
w/ wl (NTAR)	84	155	84	0.558
w/ wl (TAR)	92	170	6	0.596

Table 2. White listing results on the Dev set for FullLP.

5.4. Score Normalization Results

Table 3 contains normalization results on the Test data for the FullLP TAR condition. The row "Raw" shows the results obtained without normalization (raw posteriors). The row "ML" corresponds to the machine learning approach mentioned in Section 4.3. The normalization method improves significantly over the raw posteriors.

	ATWV
Raw	0.418
ML	0.530

Table 3. Score normalization results on the Eval set for FullLP TAR.

5.5. Semi-supervised Results

For the semi-supervised system, we first built a system using the MLP features trained on the 10-hr supervised data. This system, trained solely on supervised data, is used to transcribe the unsupervised data. Then, we performed the confidence weighted training, semi-supervised discriminative training and also the semi-supervised MLP training for the final systems, as described in Section 3. Table 4 shows the improvement of each technique in terms of WER and ATWV. All keyword search results are under the known keyword condition (TAR). The results show that semi-supervised training can improve both speech recognition and keyword search performance. Compared to the systems trained with only 10 hours of supervised data, semi-supervised training improves the system by 5.1% absolute in WER and 6.2% absolute in ATWV.

System	WER (%)	ATWV
10-hr MLP sys.	60.3	0.394
+cw training	59.0	0.408
+semi. sup. DT	58.7	0.410
+semi. sup. MLP	55.2	0.456

Table 4. Semi-supervised results on the Dev set for LimitedLP TAR.

6. CONCLUSIONS

In this paper we described the conversational telephone speech keyword spotting system under the IARPA Babel program for the 2013 evaluation. Recent advances in ASR technology are discussed including stacked bottleneck neural network features and semi-supervised training for low resource conditions. High performing KWS systems go well beyond state-of-the-art ASR technology. Important factors for KWS are including the correct answer in the lattice, via white listing, and generating meaningful scores via score normalization. Our future work will focus on variable white listing for finer control of the search and morphology and other sub-word techniques for improving OOV KWS accuracy.

7. REFERENCES

[1] D. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. Lowe, R. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Interspeech*, 2007, pp. 314–317.

[2] J. Keshet, D. Grangier, and S. Bengio, "Discriminative keyword spotting," *Speech Communication*, vol. 51, no. 4, pp. 317–329, Apr. 2009.

[3] I. Szöke, P. Schwarz, P. Matějka, and M. Karafiát, "Comparison of keyword spotting approaches for informal continuous speech," in *Eurospeech*, 2005, pp. 633–636.

[4] M. Karafiát, F. Grézl, M. Hannemann, and J. Černocký, "BUT neural network features for spontaneous Vietnamese in Babel," in *submitted to ICASSP*, 2014.

[5] J. Ma and R. Schwartz, "Unsupervised versus supervised training of acoustic models," in *Interspeech*, 2008, pp. 2374–2377.

[6] B. Zhang, R. Schwartz, S. Tsakalidis, L. Nguyen, and S. Matsoukas, "White listing and score normalization for keyword spotting of noisy speech," in *Interspeech*, 2012.

[7] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grézl, M. Hannemann, M. Karafiát, I. Szöke, K. Veselý, L. Lamel, and V.-B. Le, "Score normalization and system combination for improved keyword spotting," in *Automatic Speech Recognition and Understanding Workshop*, 2013.

[8] M. Karafiát, F. Grézl, M. Hannemann, K. Veselý, and J. H. Černocký, "BUT Babel system for spontaneous Cantonese," in *Interspeech*, 2013, pp. 2589–2593.

[9] T. Ng, B. Zhang, S. Matsoukas, and L. Nguyen, "Region Dependent Transform on MLP Features for Speech Recognition," in *Interspeech*, 2011, pp. 221–224.

[10] L. Nguyen and R. Schwartz, "Efficient 2-pass N-best decoder," in *DARPA Speech Recognition Workshop*, 1997, pp. 167–170.

[11] T. Anastasakos, J. Mcdonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *ICSLP*, 1996, pp. 1137–1140.

[12] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 4057–4060.

[13] L. Wang, M.J.F. Gales, and P.C. Woodland, "Unsupervised Training for Mandarin Broadcast News and Conversation Transcription," in *International Conference on Acoustics, Speech, and Signal Processing*, 2007, vol. 4, pp. 353–356.

[14] X. Cui, J. Huang, and J.T. Chien, "Multi-View and Multi-Objective Semi-Supervised Learning for HMM-Based Automatic Speech Recognition," *IEEE Transactions on Acoustics, Speech and Language Processing*, vol. 20, no. 7, pp. 1923–1935, 2012.

[15] B. Strope, D. Beeferman, A. Gruenstein, and X. Lei, "Unsupervised Testing Strategies for ASR," in *Interspeech*, 2011, pp. 1685–1688.

[16] K. Yu, M.J.F. Gales, L. Wang, and P.C. Woodland, "Unsupervised Training and Directed Manual Transcription for LVCSR," *Speech Communication*, vol. 52, no. 7–8, pp. 652–663, 2010.

[17] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselý, and P. Matějka, "Developing a Speech Activity Detection System for the DARPA RATS Program," in *Interspeech*, 2012.

[18] D. Povey, P.C. Woodland, and M.J.F. Gales, "Discriminative MAP for Acoustic Model Adaptation," in *International Conference on Acoustics, Speech, and Signal Processing*, 2003, vol. 1, pp. 312–315.

[19] F. Grézl and M. Karafiát, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Automatic Speech Recognition and Understanding Workshop*, 2013.

[20] I. Bulyko, O. Kimball, M.-H. Siu, J. Herrero, and D. Blum, "Detection of unseen words in conversational Mandarin," in *International Conference on Acoustics, Speech, and Signal Processing*, 2012, pp. 5181–5184.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[22] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddington, "Results of the 2006 Spoken Term Detection Evaluation," in *SIGIR 2007 Workshop Searching Spontaneous Conversational Speech*, 2007, pp. 45–50.