Joint Decoding of Tandem and Hybrid Systems for Improved Keyword Spotting on Low Resource Languages

Haipeng Wang, Anton Ragni, Mark J. F. Gales, Kate M. Knill, Philip C. Woodland, Chao Zhang

Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, UK
{hw443,ar527,mjfg,kate.knill,pcw,cz277}@eng.cam.ac.uk

Abstract

Keyword spotting (KWS) for low-resource languages has drawn increasing attention in recent years. The state-of-the-art KWS systems are based on lattices or Confusion Networks (CN) generated by Automatic Speech Recognition (ASR) systems. It has been shown that considerable KWS gains can be obtained by combining the keyword detection results from different forms of ASR systems, e.g., Tandem and Hybrid systems. This paper investigates an alternative combination scheme for KWS using joint decoding. This scheme treats a Tandem system and a Hybrid system as two separate streams, and makes a linear combination of individual acoustic model log-likelihoods. Joint decoding is more efficient as it requires just a single pass of decoding and a single pass of keyword search. Experiments on six Babel OP2 development languages show that joint decoding is capable of providing consistent gains over each individual system. Moreover, it is possible to efficiently rescore the joint decoding lattices with Tandem or Hybrid acoustic models, and further KWS gains can be obtained by merging the detection posting lists from the joint decoding lattices and rescored lattices.

Index Terms: keyword spotting, joint decoding, deep neural network, Tandem, Hybrid

1. Introduction

Keyword spotting (KWS) is the task of locating the occurrences of a given query in a large collection of audio recordings. The query can be a word or a phrase. The state-of-the-art KWS systems use automatic speech recognition (ASR) engines to generate word or sub-word lattices, and perform keyword search in these lattices. Accurate ASR training usually requires a large amount of transcribed audio data and text, which can not be satisfied for many low-resource languages. KWS for these low-resource languages is very challenging, and has drawn increasing research attention in recent years [1, 2, 3, 4]. One of the driving forces for this direction is the IARPA Babel program [5]. This program aims to facilitate rapid development of accurate KWS (the primary task) and underlying ASR technologies for any previously less studied language using a *limited* amount of training data.

With very limited training resources, a single ASR engine may not be capable of providing robust KWS performance. However, considerable gains can be obtained by combining the KWS results of multiple ASR systems. This is in line with many other information retrieval tasks, for which information fusion could significantly improve the performances [6]. Different ASR systems may start with similar features, but have significant diversity in the operation mechanisms (e,g. acoustic models and language models) that is expected to bring complementary advantages. In fact many Babel evaluation participants use system combination to improve the KWS performances [2, 7, 8]. System combination for ASR has been well established, such as ROVER combination [9] and Confusion Network Combination (CNC) [10]. This concept has also been applied to KWS. Some early work that combine systems to improve the KWS performance are [11, 12, 13], which combine KWS systems with word and sub-word models. The recent contributions in this direction include [1, 2, 7, 14, 15]. Currently most KWS combination techniques are applied on the detection posting list[1], and therefore would require a separate pass of decoding and keyword search for each individual ASR system.

This paper investigates a joint decoding scheme, which uses on-the-fly frame-level combination of the acoustic log-likelihoods of individual acoustic models. In particular, two forms of deep neural network (DNN) based acoustic models, Tandem and Hybrid, were built for joint decoding. The Tandem and Hybrid acoustic models share the same hidden Markov model (HMM) structure. In the Tandem configuration, DNN operates as a feature extractor that provides bottle-neck features for the back-end GMM-based acoustic model [16]. In the Hybrid configuration, it plays the role of the acoustic model itself and generate HMM state posteriors [17, 18, 19]. The acoustic log-likelihoods from the tandem GMM acoustic model and hybrid DNN are linearly combined as new acoustic scores, which are used in the Viterbi decoding process.

Joint decoding can be viewed as a single multi-stream system [20, 21]. Similar approaches have been investigated for ASR. In [22], the interpolation of log-likelihoods generated by a three-layer neural network and a GMM was proposed. Similar combination with a time-delay neural network and a GMM was studied in [23]. Acoustic likelihood combination and lattice combination have been compared in [24]. Joint training and decoding has also been studied in [25, 26]. This paper examines the performance of joint decoding for low-resource KWS.

This work was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract number W911NF-12-C-0012. 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. The authors would like to thank the LORELEI team for providing the KWS infrastructure and multilingual deep neural network features.

[1] Each entry in the posting list contains a query ID, occurrence time, detection score/confidence, as well as a binary Yes/No decision.

We carried out experiments in a consistent framework for two data packs on six languages. The performances of joint decoding is compared with those of individual systems as well as a posting list merging scheme. Moreover, it is efficient to rescore the joint decoding lattices with Tandem and Hybrid acoustic models. Further gains can be achieved by merging the detection posting lists from the joint decoding lattices and rescored lattices. Section 2 presents the Babel KWS task and experimental corpora. This is followed by the description of our ASR system and KWS system. Finally come the experimental results and conclusion.

2. Task Description

The work reported in this paper is based on the IARPA-funded Babel program [5]. The primary objective of this program is to foster research on technologies for rapid development of low-resource ASR and KWS systems. In each period of this program, Babel provides audio corpora for ASR system building and keyword sets for KWS tuning and evaluation. In Option Period 2 (OP2), the released development corpora involve 6 languages: Kurmanji Kurdish, Tok Pisin, Cebuano, Kazakh, Telugu, and Lithuanian. The list of official releases used in this paper is shown in Table 1.

Language	ID	Release
Kurmanji Kurdish	205	IARPA-babel205b-v1.0a
Tok Pisin	207	IARPA-babel207b-v1.0a
Cebuano	301	IARPA-babel301b-v1.0b
Kazakh	302	IARPA-babel302b-v1.0a
Telugu	303	IARPA-babel303b-v1.0a
Lithuanian	304	IARPA-babel304b-v1.0b

Table 1: Babel OP2 Languages, ID, and data releases.

For each of these languages, there are four language pack (LP) configurations, each describing a different subset of data. This paper considers two LPs: a full LP (FLP) and a very limited LP (VLLP). The approximate amount of transcribed audio, including speech and surrounding silence, is 40 hours for the FLP and 3 hours for the VLLP. The audio data comprises primarily conversational telephone speech, and is designed to contain a diverse set of speaker and recording environments. The keyword sets have quite different in-vocabulary (INV) and out-of-vocabulary (OOV) distributions across languages. For example, for the VLLP release without additional vocabulary, the OOV rate for Tok Pisin is 23.8% while the OOV rate for Telugu is 43.6%.

The system performance is measured by Maximum Term Weighted Value (MTWV) [27] for KWS and Token Error Rate (TER)[2] for ASR accuracy. Term Weighted Value (TWV) is defined as

$$TWV(\theta) = 1 - [P_{miss}(\theta) + \beta P_{fa}(\theta)] \quad (1)$$

where θ denotes the detection threshold for Yes/No decision. $P_{miss}(\theta)$ and $P_{fa}(\theta)$ denote the missing and false alarm probabilities given θ. The constant β, which is set to 999.9, decides the tradeoff between missing rate and false alarm rate. MTWV represents the maximum TWV values over the range of all possible choices of θ.

[2]TER is calculated in the same way as WER. For some languages investigated in Babel, there are only token references provided. A token could be a word, a character, a syllable, etc.

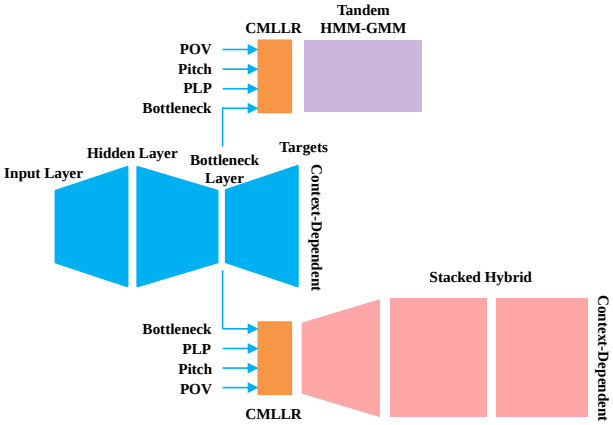

Figure 1: An illustration of Tandem and Stacked Hybrid.

3. ASR system description

The core tool for our ASR development is an extended version of the HTK toolkit [28]. The extension mainly includes a complete integration of DNNs into HTK [29].

According to OP2 program rule, the use of phonetic lexicon is excluded in system training and testing. Therefore all systems used in this paper are graphemic systems. Graphemic lexicons are generated using a unified systematic approach which is applicable to all unicode characters [30]. Global state-position based decision trees [4] are used to construct the context-dependent models. This allows unseen graphemes to be recognised, even if they do not occur in the acoustic model training data. The acoustic model training data only includes the provided transcribed data. No semi-supervised learning [31, 32, 33] approaches are adopted. Language models are estimated using the vocabulary and texts just from the acoustic transcripts.

We used two DNN based configurations, Tandem and Hybrid, for acoustic model training. This is illustrated in Figure 1. Both configurations share common front-end DNNs as feature extractors to generate bottleneck (BN) features. The front-end DNN training is initialised by layer-wise discriminative pre-training with context-independent (CI) states as targets. Due to the very limited amount of training data, the initialisation is important for VLLP, and gives about 0.5% TER reduction in a preliminary experiment with a Tandem system on Cebuano. For the FLP, the input to the front-end DNNs is 936 dimensional feature vectors, which are comprised of 9 frames of filter-bank features and pitch, appended with delta, double deltas and triple deltas. The FLP front-end is a unilingual DNN trained using the data of the target language. In contrast, the VLLP front-end is a MRASTA based multilingual DNN [34, 35], which is initially trained with the data from 11 Babel BP and OP1 FLPs and fine-tuned on the target language. The BN layer consists of 26 nodes for FLP, and 62 nodes for VLLP.

3.1. Tandem SAT

The input features for Tandem HMM-GMM systems are concatenated features, including BN features, 52-dimensional PLP+Δ+Δ^2+Δ^3, 3-dimensional pitch+Δ+Δ^2 and 3-dimensional probability of voicing (POV)+Δ+Δ^2 features. Pitch and POV are estimated using the Kaldi toolkit [36]. The final dimension of input features is 84 for FLP and 120

for VLLP. Cepstral mean normalisation (CMN) and cepstral variance normalisation (CVN) are applied. This is followed by feature transformations, including heteroscedastic linear discriminant analysis (HLDA) for PLP features, and global semi-tied transform [37] for BN features. Thus the dimensionality of Tandem features is reduced from 84 to 71 for FLP, and from 120 to 107 for VLLP. For FLP, the transformed features are then warped by Gaussianization [38].

Two sets of acoustic models are constructed. One is Speaker-Independent (SI) model, which is based on the Tandem features and estimated using Minimum Phone Error (MPE) [39] criterion. The other is estimated using Speaker Adaptive Training (SAT) [40]. SAT is performed using constrained maximum likelihood linear regression (CMLLR) [41] followed by MPE. In the decoding process, SI model with trigram LM is used to produce hypotheses for CMLLR estimation. The resulted CMLLR transforms are used to obtain speaker normalised features, which are then taken as input of the Tandem-SAT model. According to the quantity of transcribed data, the number of context-dependent (CD) states is 6000 for the FLP and 1000 for the VLLP. Each state has an average of 16 components.

3.2. Stacked Hybrid

As illustrated in Figure 1, the stacked Hybrid system use the same features as Tandem system. The same CMLLR transforms generated by Tandem-SAT models are also applied for speaker normalisation. To utilise long time span information, the input to the hybrid DNN is a concatenation of 9 consecutive feature vectors. The network configuration is different between FLP and VLLP. For FLP, a network structure of $639 \times 1000^5 \times 6000$ is used. For VLLP, the network structure is set to $963 \times 1000^4 \times 1000$. Similar to the front-end DNN training, all hybrid DNNs are initialised by layer-wise pre-training with CI targets. Fine-tuning is done using cross-entropy criterion with CD targets. The number of CD states is the same as in Tandem system. Subsequently MPE-based sequence-discriminative training [42] is applied for further improvement.

3.3. Joint Decoding

Although our Tandem system and Hybrid system use the same feature extractor, same vocabulary and language model, the differences in feature usage (1 frame vs. 9-frame concatenation) and acoustic model structure (GMM vs. DNN) are expected to provide complementary advantages. Our previous work [15] has shown that significant gains could be observed by combining these two systems. More specifically, TER reduction was obtained with CNC [10], and KWS improvement was achieved by a posting list merging scheme, which simply combined the posting lists from Tandem and Hybrid systems prior to sum-to-one (STO) normalisation [7]. Both CNC and posting list merging are post-system techniques, i.e., the combination can only be done after each individual finishes a separate complete decoding and KWS.

In contrast, this paper investigates an alternative combination scheme, which is referred to as joint decoding. As a kind of multi-stream scheme [20, 21], joint decoding treats Tandem and Hybrid as two separate streams, and makes on-the-fly combination of the acoustic log-likelihoods during one single pass of decoding. KWS can then be carried out using the joint decoding lattices. Compared to CNC and posting list merging, joint decoding is more efficient as it requires only a single pass of decoding and a single pass of keyword search. In addition, with the joint decoding lattices, it is efficient to perform rescoring with Tandem or Hybrid acoustic models. This can be done by first determining the joint decoding lattices and then decoding within the determinized lattices using one single acoustic model. Alternatively, it is also feasible to cache the arc likelihoods from each model for fast rescoring. The rescored lattices tend to be biased to one particular model, and may be complementary to the original joint decoding lattices.

During decoding, Tandem and Hybrid acoustic models are both used in acoustic observation probability computation. Given a speech frame o_t, the observation log-likelihood generated by a state s_i is computed as

$$\mathcal{L}(o_t|s_i) = \lambda_T \mathcal{L}_T(o_t|s_i) + \lambda_H \mathcal{L}_H(o_t|s_i), \quad (2)$$

where λ_T and λ_H denote the weights for Tandem and Hybrid [3], respectively. $\mathcal{L}_T(o_t|s_i)$ represents the Tandem log-likelihood calculated from the i_{th} state (assuming a GMM distribution with w_m, μ_m and Σ_m as the weight, mean and covariance for the m_{th} component, respectively),

$$\mathcal{L}_T(o_t|s_i) = \log \sum_{m \in s_i} w_m \mathcal{N}(o_t|\mu_m, \Sigma_m). \quad (3)$$

$\mathcal{L}_H(o_t|s_i)$ is the log-likelihood generated by the Hybrid model,

$$\mathcal{L}_H(o_t|s_i) = \log p(s_i|o_t) + \log p(o_t) - \log p(s_i), \quad (4)$$

where the posterior probability $p(s_i|o_t)$ is the output of the i_{th} target state in the DNN output layer, and the prior probability $p(s_i)$ is approximately estimated from the state-level alignment of the training data. $p(o_t)$ is assumed to be equal across states. The combined acoustic log-likelihoods are then used for Viterbi decoding. It is worth noting that the Tandem and Hybrid models share the same decision tree in our implementation. Due to the change in the range of acoustic log-likelihoods, the grammar scale and beam setting used in decoding should be scaled properly.

4. KWS system description

Indexing and search in our KWS implementation are based on the weighted finite state transducer (WFST) framework [14, 43]. First the audio collection is converted into word lattices with the ASR system. Word lattices are then processed to generate word index and grapheme index. These indexes contain word or grapheme identities, start and end states, and the associated posterior probabilities.

During search, a query is represented as a weighted finite state acceptor (WFSA), and subsequently the composition operation is carried out to retrieve detection postings. More specifically, each in-vocabulary (IV) query term is converted to a word WFSA, and composed with the word index. If one IV term does not get any return, it is converted to a grapheme WFSA and searched again in the grapheme index. This is known as cascade search. On the other hand, the search for out-of-vocabulary (OOV) term are operated only on the grapheme level, i.e., all OOVs are represented as grapheme WFSAs, and composed with the grapheme index. Language model scores are ignored in OOV search. To further boost the OOV detection performance, a query expansion using grapheme-to-grapheme confusability (NBestP2P) [1] is applied. NBestP2P is set to 100 in all the experiments for this paper. Finally the IV and OOV search posting lists are merged and STO score normalisation is applied to generate the final KWS output.

[3] In this paper, the same weight setting is applied to all the states for simplicity, though it is possible to design state-specific weights.

5. Experimental Results

Experiments were carried out on the development sets of six Babel OP2 languages as described in Section 2. Each language has approximately 10 hours of audio for indexing, and 2000 queries for search. The evaluation metrics are TER for ASR and MTWV for KWS. Given that the primary objective of Babel program [5] is to improve KWS performances, all the system configurations were tuned to optimise MTWV, not TER.

System	Tok Pisin		Cebuano	
	FLP	VLLP	FLP	VLLP
T	40.7	52.6	54.2	63.9
H	39.2	50.7	52.8	63.2
T $\oplus$ H	38.8	49.9	52.3	61.4
J	**38.4**	**48.6**	**52.0**	60.9
T|J	40.6	52.6	54.2	63.8
H|J	39.3	50.4	53.0	63.2
T|J $\oplus$ H|J	38.7	49.7	52.2	61.5
J $\oplus$ T|J $\oplus$ H|J	38.4	48.9	52.0	**60.7**

Table 2: ASR performance (%TER) for FLP and VLLP on Tok Pisin (207) and Cebuano (301). T: single Tandem system; H: single Hybrid system; J: Joint decoding. T|J means rescoring joint decoding lattices with Tandem models. $\oplus$ indicates CNC, i.e., T$\oplus$H indicates CNC with Tandem and Hybrid systems.

Table 2 shows the TER with different systems for Tok Pisin and Cebuano. In this table, rows 3-5 correspond to the baseline performance, while row 6-10 are all based on joint decoding. All the TER numbers are obtained with trigram LMs and CN decoding. Among the baselines, T$\oplus$H gives consistently better performances than each individual system. For joint decoding, the combination weights were tuned on Kurmanji Kurdish VLLP, and then set as $\lambda_T = 0.25$ and $\lambda_H = 1.0$. Comparing joint decoding with T$\oplus$H, better TER can be observed. This demonstrates the capability of joint decoding to exploit the complementary nature between individual models. Rescoring the joint decoding lattices gives similar performances as a separate decoding. CNC with joint decoding lattices and rescored lattices does not bring consistent TER reduction. This may indicate that lattice rescoring with a weaker acoustic model does not benefit the one-best ASR output.

System	Tok Pisin		Cebuano	
	FLP	VLLP	FLP	VLLP
T	0.4067	0.2882	0.3598	0.2254
H	0.4189	0.3132	0.3783	0.2384
T $\otimes$ H	0.4421	0.3438	0.3986	0.2711
J	0.4423	0.3444	0.4005	0.2722
T|J	0.4142	0.2939	0.3656	0.2312
H|J	0.4277	0.3247	0.3893	0.2502
T|J $\otimes$ H|J	0.4453	0.3479	0.3995	0.2736
J $\otimes$ T|J $\otimes$ H|J	**0.4476**	**0.3532**	**0.4023**	**0.2799**

Table 3: KWS performances (MTWV) for FLP and VLLP on Tok Pisin (207) and Cebuano (301). $\otimes$ represents posting list merging.

Table 3 shows the KWS performances. Among the baseline systems, T$\otimes$H, which combines the KWS posting lists prior to STO score normalisation, gives significant gains over each individual system. Joint decoding provides quite comparable MTWV performance to T$\otimes$H, with MTWV improvement ranging from 0.0002 to 0.0019. This is promising as joint decoding only requires one pass of keyword search while T$\otimes$H requires two passes. Moreover, rescoring has been applied to the joint decoding lattices with all individual acoustic models, and KWS runs have been conducted on all sets of lattices. Comparing the rescored lattices (T|J or H|J) with single decoding ones (T or H), it can be observed that the rescored lattices provide consistently better MTWV numbers. With the KWS results from all the lattices, it is interesting to see if they are complementary and can be combined via posting list merging for better performances. Corresponding performances are listed in the last line of Table 3. Consistent gains over single joint decoding systems can be observed, with MTWV improvement ranging from 0.0018 to 0.0088.

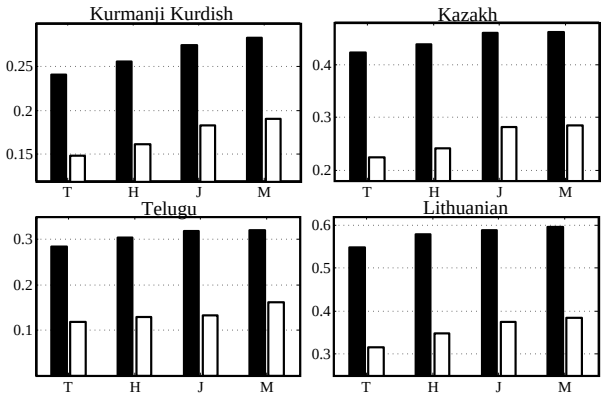

Figure 2: KWS performances on the other four Babel OP2 languages. Black bar represents FLP, and white bar represents VLLP. T: single Tandem system; H: single Hybrid system; J: Joint decoding; M: J$\otimes$ T|J $\otimes$ H|J (merging KWS posting lists of joint decoding lattices and rescored lattices).

Experiments on the other four languages were conducted in the same configurations. The KWS performance is shown in Figure 2. The advantage of joint decoding over individual systems can also be observed across all languages, with larger gains for VLLP and smaller gains for FLP. The gain from combining the KWS results of joint decoding lattices and rescored lattices is also confirmed on these four languages.

6. Conclusion

In this paper, a joint decoding scheme has been investigated for keyword spotting under the Babel program. Joint decoding combines Tandem and Hybrid systems based on the acoustic log-likelihoods. This gives information fusion of two separate systems within one single pass of decoding and keyword search. Experiments on six Babel OP2 languages have demonstrated that joint decoding can achieve comparable performances to CNC and KWS posting list merging. In addition, it is efficient to rescore the joint decoding lattices with each individual acoustic model. Further KWS gains are obtained by merging the KWS results of joint decoding lattices and rescored lattices. Although in this paper the joint decoding implementation takes in only two systems, it can be generalised to include more systems with more diverse features using a single pass of decoding. When more systems become feasible, it would be interesting to consider how to efficiently select complementary systems and optimise their combination weights.

7. References

[1] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, "Exploiting diversity for spoken term detection," in *Proc. ICASSP*, 2013, pp. 8282–8286.

[2] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, and R. Hsiao, "Score normalization and system combination for improved keyword spotting," in *Proc. ASRU*, 2013, pp. 210–215.

[3] A. Ragni, K. Knill, S. Rath, and M. Gales, "Data augmentation for low resource languages," in *Proc. Interspeech*, 2014, pp. 810–814.

[4] K. Knill, M. Gales, A. Ragni, and S. Rath, "Language independent and unsupervised acoustic models for speech recognition and keyword spotting," in *Proc. INTERSPEECH*, 2014, pp. 20–26.

[5] M. Harper, "IARPA Solicitation IARPA-BAA-11-02," 2011, http://www.iarpa.gov/solicitations_babel.html.

[6] J. Lee, "Analyses of multiple evidence combination," in *ACM SIGIR*, 1997, pp. 267–276.

[7] J. Mamou, J. Cui, X. Cui, M. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran *et al.*, "System combination and score normalization for spoken term detection," in *Proc. ICASSP*, 2013, pp. 8272–8276.

[8] N. Chen *et al.*, "Low-resource keyword search strategies for TAMIL," in *Proc. ICASSP*, 2015, pp. 5366–5370.

[9] J. G. Fiscus, "A Post-processing System to Yield Reduced Word Error Rates: Recogniser Output Voting Error Reduction (ROVER)," in *Proc. ASRU*, 1997, pp. 347–354.

[10] G. Evermann and P. Woodland, "Posterior Probability Decoding, Confidence Estimation and System Combination," in *Proc. Speech Transcription Workshop*, vol. 27, 2000.

[11] 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 *Proc. Interspeech*, 2007.

[12] D. Vergyri, I. Shafran, A. Stolcke, V. Gadde, M.Akbacak *et al.*, "The SRI/OGI 2006 spoken term detection system," in *Proc. Interspeech*, 2007, pp. 2393–2396.

[13] I. Szoke, L. Burget, J. Cernocky, and M. Fapso, "Sub-word modeling of out of vocabulary words in spoken term detection," in *Proc. SLT*, 2008, pp. 273–276.

[14] B. Kingsbury *et al.*, "A high-performance Cantonese keyword search system," in *Proc. ICASSP*, 2013, pp. 8277–8281.

[15] S. Rath, K. Knill, A. Ragni, and M. Gales, "Combining tandem and hybrid systems for improved speech recognition and keyword spotting on low resource languages," in *Proc. Interspeech*, 2014, pp. 835–839.

[16] J. Park, F. Diehl, M. Gales, M. Tomalin, and P. C. Woodland, "The efficient incorporation of MLP features into automatic speech recognition systems," *Computer Speech and Language*, vol. 25, no. 3, pp. 519–534, 2010.

[17] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[18] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov 2012.

[19] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: a Hybrid Approach*. Springer Science & Business Media, 1994, vol. 247.

[20] A. Morris, A. Hagen, H. Glotin, and H. Bourlard, "Multi-stream adaptive evidence combination for noise robust ASR," *Speech Communication*, vol. 34, no. 1, pp. 25–40, 2001.

[21] H. Misra, H. Bourlard, and V. Tyagi, "New entropy based combination rules in HMM/ANN multi-stream ASR," in *Proc. ICASSP*, 2003, pp. 738–741.

[22] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in hmm speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 1, pp. 161–174, 1994.

[23] C. Dugast, L. Devillers, and X. Aubert, "Combining TDNN and HMM in a hybrid system for improved continuous-speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 1, pp. 217–223, 1994.

[24] P. Swietojanski, A. Ghoshal, and S. Renals, "Revisiting hybrid and gmm-hmm system combination techniques," in *Proc. ICASSP*, 2013, pp. 6744–6748.

[25] X. He and K. Toutanova, "Joint optimization for machine translation system combination," in *Proc. EMNLP*, 2009, pp. 1202–1211.

[26] H. Soltau, G. Saon, and T. Sainath, "Joint training of convolutional and non-convolutional neural networks," *Proc. ICASSP*, 2014.

[27] J. Fiscus, J. Ajot, J. Garofolo, and G. Doddingtion, "Results of the 2006 Spoken Term Detection Evaluation," in *Proc. SIGIR*, 2007, pp. 51–57.

[28] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, *The HTK Book (for HTK version 3.4.1)*. `http://htk.eng.cam.ac.uk`: Cambridge University, 2009.

[29] C. Zhang and P. Woodland, "A general artificial neural network extension for HTK," in *Submission to InterSpeech*, 2015.

[30] M. Gales, K. Knill, and A. Ragni, "Unicode-based graphemic systems for limited resource languages," in *Proc. ICASSP*, 2015.

[31] L. Lamel and J.-L. Gauvain, "Lightly supervised and unsupervised acoustic model training," *Computer speech and language*, vol. 16, pp. 115–129, 2013.

[32] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Proc. ICASSP*, 2013, pp. 6704–6708.

[33] R. Hsiao, T. Ng, F. Grézl, D. Karakos, S. Tsakalidis, L. Nguyen, and R. Schwartz, "Discriminative semi-supervised training for keyword search in low resource languages," in *Proc. ASRU*, 2013, pp. 440–445.

[34] Z. Tuske, J. Pinto, D. Willett, and R. Schluter, "Investigation on cross- and multilingual MLP features under matched and mismatched acoustical conditions," in *Proc. ICASSP*, 2013, pp. 6970–6974.

[35] Z. Tuske, D. Nolden, R. Schluter, and H. Ney, "Multilingual MRASTA features for low-resource keyword search and speech recognition systems," in *Proc. ICASSP*, 2014, pp. 7854–7858.

[36] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[37] M. Gales, "Semi-tied covariance matrices for hidden markov models," *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 3, pp. 272–281, 1999.

[38] G. Saon, S. Dharanipragada, and D. Povey, "Feature space gaussianization," in *Proc. ICASSP*, 2004, p. 326329.

[39] D. Povey and P. C. Woodland, "Minimum Phone Error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, 2002, pp. 101–105.

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

[41] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[42] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Proc. Interspeech*, 2013, pp. 2345–2349.

[43] M. Mohri, C. Allauzen, and M. Saraclar, "General indexation of weighted automata – application to spoken utterance retrieval," in *Proc. HLT/NAACL*, 2004, pp. 33–40.