SYSTEM COMBINATION AND SCORE NORMALIZATION FOR SPOKEN TERM DETECTION

Jonathan Mamou[1], *Jia Cui*[2], *Xiaodong Cui*[2], *Mark J. F. Gales*[3], *Brian Kingsbury*[2], *Kate Knill*[3], *Lidia Mangu*[2], *David Nolden*[4], *Michael Picheny*[2], *Bhuvana Ramabhadran*[2], *Ralf Schlüter*[4], *Abhinav Sethy*[2], *Philip C. Woodland*[3]

[1]IBM Haifa Research Labs, Haifa 31905, Israel
[2]IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
[3]Cambridge University Engineering Department, Trumpington St., Cambridge CB2 1PZ, U.K.
[4]Chair of Computer Science 6, RWTH Aachen University, Ahornstr. 55, D-52056 Aachen, Germany

ABSTRACT

Spoken content in languages of emerging importance needs to be searchable to provide access to the underlying information. In this paper, we investigate the problem of extending data fusion methodologies from Information Retrieval for Spoken Term Detection on low-resource languages in the framework of the IARPA Babel program. We describe a number of alternative methods improving keyword search performance. We apply these methods to Cantonese, a language that presents some new issues in terms of reduced resources and shorter query lengths. First, we show score normalization methodology that improves in average by 20% keyword search performance. Second, we show that properly combining the outputs of diverse ASR systems performs 14% better than the best normalized ASR system.

Index Terms— spoken term detection, keyword search, data fusion, system combination, score normalization

1. INTRODUCTION

The rapidly increasing amount of spoken data calls for solutions to index and search this data. In 2006, the U.S. National Institute of Standards and Technology (NIST) created the STD (Spoken Term Detection) evaluation [1] initiative to facilitate research and development of technology for retrieving information from archives of speech data. However, this work was performed on large-resource languages and most of the effort focused on clean speech. It has recently been demonstrated that significant improvement on STD task can be obtained by deliberately designing diverse and complementary ASR components (i.e., front ends, acoustic models, etc) [2]. We show that similar approach works on noisy speech for lowresource languages with low target false alarm rate. The contribution of the paper is twofold. First, we show the value of combining results from diverse ASR systems that employ different front ends in order to provide reliable Keyword Search (KWS) results, using and extending state-of-the-art Information Retrieval (IR) data fusion methodologies. Techniques for data fusion are used extensively in many different application areas, and IR is one of them. By combining results from diverse ASR systems, we show good robustness across a wide variety of talkers, channels, environments, and target terms. Second, we compare score normalization approaches for STD. The score normalization is relevant to data fusion since those scores provided by the different systems are not comparable. Therefore, score normalization is often performed as a preliminary step to data fusion. In our context, the scores are output by ASR systems and represent posterior probability estimates. For this reason, for STD, the normalization is not necessarily required as a preprocessing step. However, according to the metric used for the task, the Actual Term-Weighted Value (ATWV), the benefit of correctly finding a term is inversely proportional to the term frequency while the cost of a false alarm is almost independent to the term frequency. In other words, ATWV metric emphasizes recall of rare terms. For this reason, score normalization may contribute to the optimization of the search performance of a single system independently of the system combination. In this paper, we investigate different score normalization and system combination methodologies and show their impact on ATWV metric. The basic processing flow is as follows. The audio data is transcribed using diverse ASR systems. Each ASR system output is indexed separately. Each query is searched against the different indices. The scores of the hits are possibly normalized. The hit lists returned by the different indices are merged to form a single *meta-hit* list for the query and a score is attributed to the meta-hit. The paper is organized as follows. After presenting the task (Section 2), we describe our KWS approach (Section 3). Next, we present score normalization methods (Section 4) and system combination approaches (Section 5). We relate these methodologies to prior work (Section 6). Experiments and analysis are presented on Cantonese spoken data (Section 7). Finally, we conclude (Section 8).

2. TASK DESCRIPTION

The present work addresses the STD task defined by NIST for the 2006 STD Evaluation with some modifications introduced by IARPA's Babel program [3]. The task consists in finding all the exact matches of a specific query in a given corpus of speech data. A query is a textual phrase containing one or several terms. We focus in the task where the system components and word indices are frozen before the queries are provided. KWS performance is measured by the ATWV metric, which combines missed detection and false alarm error types [1]. More precisely, Term-Weighted Value (TWV) is 1 minus the weighted sum of the term-weighted probability of missed detection and the term-weighted probability of false alarms. ATWV is the TWV attained by the system as a result of the system output and the binary decision output for each putative occurrence assigned according to a global detection threshold. MTWV is the maximum TWV over the range of all possible values of the detection threshold.

3. KEYWORD SEARCH

In this section, we describe our KWS system; it exploits the flexibility of a weighted finite state transducer (WFST) based indexing system [4, 5]. The indexing and search components are presented. Note that the same indexing and search process are run for each ASR system independently.

Indexing: we assume that the audio to be indexed has been processed with a large vocabulary continuous speech recognition system and the corresponding word lattices are available. Phonetic lattices are subsequently derived from these word lattices and are used to build phonetic indices for out-of-vocabulary (OOV) search. The timing information is pushed onto the output label of each arc in the lattice. Let denote Q as a finite set of states. Every arc in the resulting WFST representing the lattice is a 5-tuple (p, i, o, w, q) where $p \in Q$ is the start state, $q \in Q$ is the end state, i is the input label (for example, a phone), o is the output label (start-time associated with state p), and w is (negative logarithm of) posterior probability associated with i. All silences and hesitations in the lattices are converted to ϵ arcs in the WFST representation.

Search: each query is represented as a weighted acceptor. A composition operation [6] with the index retrieves the lattice (utterance) containing it. Each query is split into two categories: in and out-of-vocabulary terms. The in-vocabulary (IV) terms are searched through the word index and the OOV terms are searched through the phonetic index derived from the word lattices. At query time, an orthographic representation of the query is converted to a sensible phonetic representation. This is typically done using grapheme-to-phoneme conversion algorithms which may not work accurately for all query terms. For Cantonese, which is an ideographic language, we use a rule based approach to generate pronunciations. The two hit lists (IV and OOV) are joined to generate the final hit list where each hit is identified by its audio file, begin time, duration and score (posterior probability estimate). The hit scores are possibly normalized as described in the next section.

4. SCORE NORMALIZATION METHODOLOGIES

In this section, we investigate different score normalization methodologies that contribute to increase ATWV. This value is directly affected by missed detections and false alarms. According to ATWV definition, the cost of a missed detection of a query is inversely proportional to its frequency while the cost of a false alarm is almost independent to its frequency. Score normalization methods have been studied for data fusion in IR [7] and we present three methodologies that improve KWS performance. Suppose that there are N_q hits for the query q according to a given ASR system. Let $s_{q,i}$ denote the score of the i-th hit for the query q.

Query Length Normalization (QL): in order to further reduce false alarms, we incorporate a *query length* (**QL**) normalization based on the duration of the hits of the query. Since hits with a longer duration are more likely to be correct, we define the QL normalization as:

$$s_{q,i}^{\frac{1}{\Delta_{avg(q)}}}$$

where $\Delta_{avg(q)}$ is the average duration of all returned hits for the query q. This approach is inspired from [8].

Sum-to-one Normalization (STO): *sum-to-one* normalization computes new scores as

$$\frac{s_{q,i}}{\sum_{j=1}^{N_q} s_{q,j}}$$

For a given query, the sum of all the normalized hit scores is 1.0. For the special case of a single hit, the normalized score is 1.0 by definition. This normalization scheme was proposed for IR data fusion in [9] and showed improvement for meta-search. It was used successfully for the first time in STD in [2]. A variant of scheme was initially investigated for IR in [10]. For STD, the denominator is the sum of the posterior estimates for all the hits; it represents an approximation of the number of occurrences of the query. For rare terms, the denominator will be low and therefore the normalized score will be high and is likely to be above the decision threshold; therefore, the probability of missed detection will be lower.

Regression-based Normalization (Pace): we describe a machine learning procedure for score normalization. We use the pace regression algorithm [11] to learn a scoring function that combines the following six features of a hit: its posterior probability, the number of occurrences of the query (it is approximated by the sum of the posterior estimates for all the hits of the query), the duration of the hit, the average duration of all the returned hits for the query, the number of words and the number of characters in the query.

5. SYSTEM COMBINATION METHODOLOGIES

In this section, we investigate system combination methodologies that contribute to increase KWS performance. Data fusion methodologies are widely used for document IR [9]. For example, in the context of meta-search engines, a hit is identified by a document and a score. Hits returned by different engines may be merged into a single meta-hit if they refer to the same document. Determining the score of the meta-hit is a key issue. Several fusion methods for combining multiple scores for document retrieval have been proposed. Some methods select one extreme end of the sample values to be the representative score of a document in the fused results (e.g., CombMIN and CombMAX), whereas the others use some form of the sum of all sample values as the final score (e.g., CombSUM, CombANZ, CombMNZ). Some methods also emphasize or de-emphasize those documents that appear multiple times in the different results (e.g., CombANZ and CombMNZ). Experimental studies have been conducted on these methods [12]; the conclusion is that CombSUM and CombMNZ are the best among the five methods for document retrieval. Let denote h_i the hit that contributes to the meta-hit H from the i-th system among n systems, and $s(h_i)$ the score of the hit h_i. More precisely, CombSUM consists of computing the summation of the hit scores from the different indices and is computed as $\sum_{i=1}^{n} s(h_i)$; CombMNZ consists of computing the summation of the hit scores from the different indices and to multiply it by the number of indices having a non-zero score for this meta-hit; it is computed as $m_H \times \sum_{i=1}^{n} s(h_i)$ where m_H is the number of indices having a non-zero score for this meta-hit H. A natural extension of CombSUM is the linear combination method where each component system i is assigned a weight w_i. It is computed as $\sum_{i=1}^{n} w_i \cdot s(h_i)$. Determining the weights is a key issue. Linear regression is generally used to assign the weights. Another weighting policy is to associate weights with performance. For STD, a hit is identified by its audio file, begin time, duration and score. Hits returned by the diverse ASR systems are merged to form a single meta-hit if they overlap in time. We identify the meta-hit by the audio file, the start time and duration taken from the highest-scoring hit. KWS performance of a system is estimated by its MTWV and ATWV. Therefore, we propose the *MTWV-weighted CombMNZ* (**WCombMNZ**) fusion methodology, which extends CombMNZ by incorporating MTWV-based weighting. For each ASR system, the hit scores from that system are weighted in proportion to the MTWV of that system for

some tuning data. Let us denote $MTWV_i$ the $MTWV$ of the i-th system. The MTWV based weight on the i-th system, w_i^{MTWV} is defined by

$$\frac{MTWV_i}{\sum_{j=1}^{n} MTWV_j}$$

For WCombMNZ, we define the score of the meta-hit H as

$$m_H \times \sum_{i=1}^{n} w_i^{MTWV} \cdot s(h_i)$$

.

6. RELATED WORK

Traditional STD approaches propose to combine word and phonetic search (e.g., [8, 13, 14, 15, 16]); phonetic lattices are included in the search in order to overcome the OOV issue. However, a query term is searched against a single index according to its type (IV or OOV) and the combination is not achieved at the query hit list level. A term-specific detection threshold derived by maximizing the expected value of ATWV per query is proposed by [14]. However, in the Babel task, we are required to work with a global detection threshold. We can easily re-normalize the term-specific threshold to a global threshold, e.g., by dividing the score by its term-specific threshold. However, this approach seems less intuitive and does not provide any gain comparing with STO normalization. In [17], they propose to normalize scores by replacing each score sc with $1 - pFA(sc)$; they present a methodology to estimate $pFA(sc)$ on a tuning set. We compare this approach to our methodologies in Section 7.2. Combination of hit lists has been used for spoken document retrieval in order to combine search resuls from transcripts that are produced according to different word and sub-word decoding methods in order to solve the OOV issue [18, 19]; the improvement is measured by the precision and the recall of the KWS. The work presented here shows that the combination strategy is useful on a very different task with very different challenges. Recently, it has been demonstrated, as part of the DARPA RATS program, that good KWS performance on STD can be obtained by combining ASR systems [2]. However, in the RATS task, the main challenges are severe noise and channel distortion, while in the Babel task, the main challenges are speaker variability and severely limited LM training data. Moreover, to our knowledge, our paper is the first work investigating state-of-the-art IR data fusion approaches for STD.

7. EXPERIMENTS

7.1. Experimental Setup

Results are reported using the above score normalization and system combination methods on the Cantonese language collection from the IARPA Babel Program (release babel101b-v0.4c); it corresponds to the data for the "dry run" August 2012 evaluation. The data collection covers multiple aspects of Cantonese spanning dialects, topics, gender and age. The training data is basically telephone conversational data and some scripted data. It contains 200 hours of training audio; approximately 50% of the training data is silence. The test data is limited to only conversational data. The query set includes 1000 queries, and are searched against the development data that is split into a tuning set (13 hours of audio) and a validation set (7 hours of audio). The repartition of the queries according to query type (IV, OOV) and to query length (measured by the number of characters) is respectively provided in Table 5 and in Table 4. We consider a query as OOV if it contains at least one OOV term. The KWS results are produced for six different ASR systems: (1) **GMM**, the baseline GMM/HMM system which is a discriminatively trained, speaker-adaptively trained acoustic model; (2) **BSRS**, a Bootstrap and restructuring model [20] in which the original training data is randomly re-sampled to produce multiple subsets and the resulting models are aggregated at the state level to produce a large, composite model; (3) **CU-HTK**, a TANDEM HMM system from Cambridge University using cross-word, state-clustered, triphone models trained with MPE, fMPE, and speaker-adaptive training. For efficiency, the MLP features were incorporated in the same fashion as [21]; (4) **MLP**, a multi-layer perceptron model [22] which is a GMM-based ASR system that uses neural-network features; (5) **NN-GMM**, a speaker-adaptively and discriminatively trained GMM/HMM system from RWTH Aachen University using bottle-neck neural network features [23] and a 4-gram Kneser-Ney LM with optimized discounting parameters [24] using a modified version of the RWTH open source decoder [25]; and (6) **DBN**, a deep belief network hybrid model [26, 27] with discriminative pertraining, frame-level cross-entropy training and state-level minimum Bayes risk sequence training. GMM, BSRS, DBN and MLP models are built with the IBM Attila toolkit [28]. A 3-gram LM with modified Kneser-Ney smoothing [29] is applied for these models. The ASR systems are described in more details in [30]. Our KWS system is implemented using the OpenFst toolkit [31]. ATWV and MTWV are evaluated using the F4DE NIST Evaluation tool [32]. ATWV results for the KWS on the validation set are computed using the optimal detection threshold obtained for the KWS on the tuning set. If necessary, the decision threshold is rescaled to take into account the size difference between the tuning and the evaluation sets. We report the Character Error Rate (CER) and the lattice density (in arcs per second of audio) of each ASR system on the development data in Table 1.

Model	% CER	lattice density
GMM	55.9	678
BSRS	53.0	1196
CU-HTK	52.9	4123
MLP	52.8	611
NN-GMM	52.7	2100
DBN	48.9	1224

Table 1. Performance of six ASR systems measured in terms of CER and lattice density.

7.2. Score Normalization

We compare KWS performance on the validation set for different normalization methodologies and we present the results in Table 2. In the baseline column, we report ATWV results with un-normalized raw scores (posterior probabilities). We denote the approach proposed by [17] and described in Section 6 as **pFA**; pFA values are estimated on the tuning set. In contrast to pFA and regression-based approaches, STO and QL methods does not rely on learning the scoring on a tuning set. Best KWS performance is obtained for DBN ASR system with STO normalization methodology. We observe that, in average, STO method provides a relative improvement of 20% over the baseline, while the relative improvement is respectively 7%, 14% and 15% for QL, Pace and pFA methods. Note that higher CER can sometimes lead to higher KWS performance; it is explained by the fact that the reported CERs are computed on the 1-best consensus

Model	baseline	QL	pFA	Pace	STO
GMM	0.300	0.345	0.335	0.360	0.375
BSRS	0.392	0.402	0.417	0.415	0.444
CU-HTK	0.368	0.410	0.443	0.434	0.453
MLP	0.362	0.381	0.410	0.407	0.416
NN-GMM	0.360	0.387	0.445	0.430	0.465
DBN	0.423	0.442	0.467	0.469	0.483

Table 2. ATWV results for six ASR systems according to different score normalization methodologies.

Methodology	ATWV
STO-LC	0.512
STO-CombMNZ	0.536
STO-WCombMNZ	0.540
WCombMNZ-STO	0.544
STO-WCombMNZ-STO	0.551

Table 3. ATWV results according to different system combination methodologies.

network while the queries are searched on the whole lattices.

7.3. System Combination

We compare KWS performance according to different system combination methodologies presented in Section 5. In IR, score normalization is applied before the data fusion in order to make the scores from the different systems comparable. However, in STD, it is not necessary since scores represent posterior probability estimates. Moreover, it can affect KWS performance in some cases, especially if the different ASR systems produce lattices having different densities. We present in Table 3 the ATWV results on the validation set according to the following system combination methodologies: (1) **STO-LC**, STO normalized scores are combined according to the linear combination methodology; the weights are determined using the multinomial logistic regression model [33] on the tuning set; (2) **STO-CombMNZ**, STO normalized scores are combined according to CombMNZ methodology; (3) **STO-WCombMNZ**, STO normalized scores are combined according to WCombMNZ methodology; (4) **WCombMNZ-STO**, raw scores are combined according to WCombMNZ methodology; and (5) **STO-WCombMNZ-STO**, STO normalized scores are combined according to WCombMNZ methodology, and after the combination, the scores of the metahits are normalized according to STO. The MTWV-based weights of WCombMNZ approaches are estimated on the tuning set. We observe that the WCombMNZ approaches outperform the traditional data fusion approaches (STO-LC and STO-CombMNZ) and provide a relative improvement of up to 14% over the best normalized system (DBN with STO normalization).

7.4. Analysis

We analyze the effect of STO normalization and system combination on KWS performance, both as a function of query length in characters and a function of query type (IV or OOV). Theokse measurements are made on the development data. Table 4 presents a breakdown of results by query type. For each type, we provide the rate of queries belonging to this type. Table 5 presents a breakdown of results by query length. For each length, we provide the rate of queries having this length and the rate of OOV queries among the queries of this length. For both tables, we report respectively in the last columns the relative improvement of STO normalization over the baseline for the best ASR system (DBN) and the relative improvement of STO-WCombMNZ-STO system combination over the best normalized single ASR system (DBN with STO normalization). We can draw a number of conclusions from these tables: (1) STO normalization and STO-WCombMNZ-STO system combination improve ATWV for both IV and OOV queries, with the effect being strongest for OOV queries; (2) STO normalization improves ATWV for all query lengths, with the effect being strongest for the shortest queries; (3) STO-WCombMNZ-STO system combination improves ATWV for all query lengths, with the effect being strongest for the longer queries; and (4) the strong system combination improvement for 4-character queries is due to both of these effects (long queries with high OOV rate).

query type	% queries	% STO improv.	% combination improv.
IV	87	15	11
OOV	13	94	69

Table 4. STO normalization and STO-WCombMNZ-STO combination ATWV relative improvement as a function of query type.

query length	% queries	% OOV	% STO improv.	% combination improv.
2	39	12	14	5
3	43	13	17	16
4	14	19	8	23
5	4	15	7	18

Table 5. STO normalization and STO-WCombMNZ-STO combination ATWV relative improvement as a function of query length.

8. CONCLUSION

In this paper, we show the STO score normalization methodology that improves in average by 20% KWS performance, achieving an ATWV of 0.483 for DBN system, and the STO-WCombMNZ-STO system combination approach that improves by 14% KWS performance, achieving an ATWV of 0.551. As future work, we will apply these approaches to other low-resource languages in the framework of Babel program; we will investigate discriminative system combination algorithms including query and system specific features, and other features like prosody and rich linguistic context information.

9. ACKNOWLEDGMENT

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract number W911NF-12-C-0012. [1] The authors are grateful to Janice Kim of IBM Research for providing software support, and to Roger Hsiao of BBN and the Babelon team, for sharing their partition of babel101b-v0.4c development data into tuning and validation sets.

[1]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.

10. REFERENCES

[1] J.G. Fiscus, J. Ajot, J.S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proceedings of ACM SIGIR Workshop on Searching Spontaneous Conversational*. Citeseer, 2007, pp. 51–55.

[2] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, "Exploiting diversity for spoken term detection," in *Proc. ICASSP*, 2013. To appear.

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

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

[5] D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraclar, "Effect of pronounciations on oov queries in spoken term detection," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3957–3960.

[6] M. Mohri, F. Pereira, and M. Riley, "Weighted automata in text and speech processing," *arXiv preprint cs/0503077*, 2005.

[7] S. Wu, F. Crestani, and Y. Bi, "Evaluating score normalization methods in data fusion," *Information Retrieval Technology*, pp. 642–648, 2006.

[8] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proceedings of the 30 th annual international ACM SIGIR conference*, 2007, vol. 23, pp. 615–622.

[9] S. Wu, *Data Fusion in Information Retrieval*, vol. 13, Springer, 2012.

[10] M. Montague and J.A. Aslam, "Relevance score normalization for metasearch," in *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 2001, pp. 427–433.

[11] Y. Wang and I.H. Witten, "Pace regression," 1999.

[12] J.H. Lee, "Analyses of multiple evidence combination," in *ACM SIGIR Forum*. ACM, 1997, vol. 31, pp. 267–276.

[13] D. Vergyri, I. Shafran, A. Stolcke, R.R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The sri/ogi 2006 spoken term detection system," in *Proc. Interspeech*. Citeseer, 2007, vol. 7, pp. 2393–2396.

[14] D.R.H. Miller, M. Kleber, C. Kao, O. Kimball, T. Colthurst, S.A. Lowe, R.M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007, vol. 7, pp. 314–317.

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

[16] B. Ramabhadran, A. Sethy, J. Mamou, B. Kingsbury, and U. Chaudhari, "Fast decoding for open vocabulary spoken term detection," in *Proceedings of HLT: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Association for Computational Linguistics, 2009, pp. 277–280.

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

[18] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," *Urbana*, vol. 51, pp. 61801, 2004.

[19] J. Mamou, Y. Mass, B. Ramabhadran, and B. Sznajder, "Combination of multiple speech transcription methods for vocabulary independent search," in *Searching Spontaneous Conversational Speech Workshop, SIGIR*, 2008, pp. 20–27.

[20] X. Cui, J. Xue, X. Chen, P. A. Olsen, P. L. Dognin, U. V. Chaudhari, J. R. Hershey, and B. Zhou, "Hidden markov acoustic modeling with bootstrap and restructuring for low-resourced languages," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 20, no. 8, pp. 2252–2264, 2012.

[21] J. Park, F. Diehl, MJF Gales, M. Tomalin, and PC Woodland, "The efficient incorporation of mlp features into automatic speech recognition systems," *Computer Speech & Language*, vol. 25, no. 3, pp. 519–534, 2011.

[22] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. ICASSP*, 2012.

[23] M. Nussbaum-Thom, Z. Tüske, G. Heigold, R. Schlüter, and H. Ney, "Posterior-scaled mpe: Novel discriminative training criteria," in *Proc. Interspeech*, 2012.

[24] M. Sundermeyer, R. Schlüter, and H. Ney, "On the estimation of discount parameters for language model smoothing," *Interspeech, Florence, Italy*, 2011.

[25] D. Nolden, D. Rybach, H. Ney, et al., "Joining advantages of word-conditioned and token-passing decoding," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4425–4428.

[26] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Improving training time of deep belief networks through hybrid pre-training and larger batch sizes," in *Proc. NIPS Workshop on Log-linear Models*, 2012.

[27] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum bayes risk training of deep neural network acoustic models using distributed hessian-free optimization," in *Proc. Interspeech*, 2012.

[28] H. Soltau, G. Saon, and B. Kingsbury, "The ibmattila speech recognition toolkit," in *Proc. IEEEWorkshop on Spoken Language Technology*, 2010.

[29] S. F. Chen and J. T. Goodman, "An empirical study of smoothing techniques for language modeling," in *In Proceedings of the 34th ACL*, 1996, pp. 310–318.

[30] B. Kingsbury, J. Cui, X. Cui, M. J. F. Gales, K. Knill, J. Mamou, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlüter, A. Sethy, and P. C. Woodland, "A high-performance Cantonese keyword search system," in *Proc. ICASSP*, 2013. To appear.

[31] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "Openfst: A general and efficient weighted finite-state transducer library," *Implementation and Application of Automata*, pp. 11–23, 2007.

[32] "NIST Tools," `http://www.itl.nist.gov/iad/mig/tools/`.

[33] S. Le Cessie and JC Van Houwelingen, "Ridge estimators in logistic regression," *Applied statistics*, pp. 191–201, 1992.