A HIGH-PERFORMANCE CANTONESE KEYWORD SEARCH SYSTEM

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

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

ABSTRACT

We present a system for keyword search on Cantonese conversational telephony audio, collected for the IARPA Babel program, that achieves good performance by combining postings lists produced by diverse speech recognition systems from three different research groups. We describe the keyword search task, the data on which the work was done, four different speech recognition systems, and our approach to system combination for keyword search. We show that the combination of four systems outperforms the best single system by 7%, achieving an actual term-weighted value of 0.517.

Index Terms— keyword search, spoken term detection, system combination, deep learning

1. INTRODUCTION

Keyword search, also known as spoken term detection, is a speech processing task in which the goal is to find all occurrences of a word or consecutive sequence of words (a "term"), presented in orthographic form, in a large audio corpus. Of particular interest is pre-indexed keyword search, in which the corpus to be searched is indexed without knowledge of the query terms. Because they do not access the original audio, pre-indexed systems are expected to have better response times for interactive search. The pre-indexed constraint distinguishes keyword search from previous work on keyword spotting [1] in which explicit models for keywords and fillers are constructed before the audio is decoded. Unlike keyword spotting systems, keyword search systems must cope with the case where one or more words in a search term were not present in the vocabulary of the speech recognition system used to index the corpus.

Like many other areas in human language technology, research on keyword search has been substantially advanced by competitive evaluations. The first evaluation of keyword search was the STD 2006 evaluation [2, 3, 4], a pilot competition run by the U.S. National Institute of Standards and Technology (NIST) in 2006. The STD 2006 evaluation measured keyword search performance on three different languages, English, Arabic, and Mandarin Chinese, and three different genres, broadcast news (BN), conversational telephony speech (CTS), and meetings. Note that only English meeting data were available, and that for the Arabic tasks the broadcast news material was Modern Standard Arabic, while the conversational telephony material was Levantine Arabic. A key contribution of the STD 2006 evaluation was the development by NIST of a metric for keyword search: actual term-weighted value, which is described in Section 2.

The STD 2006 evaluation also revealed a close relationship between the performance of keyword search and the state of the art for speech recognition on a given combination of language and genre. The best performance was obtained on English broadcast news and conversational telephony, two very well studied tasks with substantial resources available for model training. Much worse performance was achieved on Arabic and Chinese conversational telephony data and on English meeting data, tasks that have fewer resources and that have attracted less research focus. While it is not possible to directly compare keyword search performance across languages and genres due to the strong dependence of keyword search on query properties such as length and frequency, the differences described here are large enough to be considered reliable, with scores on the English BN and CTS being more than double those achieved on the Arabic CTS, Chinese CTS, and English meetings data.

Reducing the performance gap between high-resource, well-studied languages and low-resource, lightly studied languages is one of the primary aims of the IARPA Babel program: "[t]he goal of the Babel Program is to be able to rapidly develop speech recognition capability for keyword search in a previously unstudied language, working with speech recorded in a variety of conditions with limited amounts of transcription." [5]. To achieve this goal, in each approximately one-year period of the program, participants work with a diverse set of development languages to gain experience with keyword search. In the first period, these are Cantonese, Tagalog, Pashto and Turkish. At the end of each period, there are evaluations of keyword search performance on the development languages, and on a previously unseen *surprise* language where participants have a limited period of time for system development.

In August 2012 there was a "dry run" evaluation on Cantonese keyword search to exercise the evaluation infrastructure and give teams a chance to practice running an evaluation. In this paper we describe a Cantonese keyword search system that was assembled after this dry run, and incorporated a number of lessons learned during and following the dry run. After explaining the actual term weighted value metric (Section 2), we describe the Cantonese training and development data used in this work (Section 3). Next, we discuss the architecture of our keyword search system (Section 4), the speech recognition systems that are used to index the audio (Section 5), the weighted finite state transducer approach to keyword search (Section 6), and our approach to score normalization and system combination (Section 7). Finally, we relate this keyword search system to prior work (Section 8) and then describe the performance of the component systems and combined keyword search system on the Babel Cantonese development data (Section 9).

2. MEASURING KEYWORD SEARCH PERFORMANCE

Keyword search is fundamentally a detection task: given an audio corpus and a list of queries, the system returns a list of possible occurrences of the terms, where each entry in the list contains (1) the detected query; (2) the start and end time of the query occurrence; (3) a binary YES/NO decision on whether the keyword search system considers the occurrence to be correct; and (4) a detection score indicating system confidence; with higher scores corresponding to greater confidence. Borrowing terminology from information retrieval, we refer to such a list as a "postings list." The YES/NO decision and detection score are not necessary in an operational system, but are useful for measuring performance.

To score a postings list for a given detection threshold θ, entries in the list are matched to reference occurrences using an objective function that accounts for both temporal overlap between the reference and putative occurrences and the detection scores assigned by the system. Following matching, for each term $\mathcal{T}$ in query list $\mathcal{Q}$, the probabilities of miss and false alarm errors are computed as

$$P_{\text{Miss}}(\mathcal{T},\theta) = 1 - N_{\text{correct}}(\mathcal{T},\theta)/N_{\text{ref}}(\mathcal{T}) \quad (1)$$

$$P_{\text{FA}}(\mathcal{T},\theta) = N_{\text{spurious}}(\mathcal{T},\theta)/N_{\text{trial}}(\mathcal{T}) \quad (2)$$

where $N_{\text{ref}}(\mathcal{T})$ is the number of reference occurrences of $\mathcal{T}$, $N_{\text{correct}}(\mathcal{T},\theta)$ is the number of correctly hypothesized occurrences of $\mathcal{T}$ at detection threshold θ, $N_{\text{spurious}}(\mathcal{T},\theta)$ is the number of incorrectly hypothesized occurrences of $\mathcal{T}$ at threshold θ, and $N_{\text{trial}}(\mathcal{T})$ is the number of trials for $\mathcal{T}$. Because a collection of continuous audio streams is processed, there is no discrete set of trials; therefore, $N_{\text{trial}}(\mathcal{T})$ is given the somewhat arbitrary definition

$$N_{\text{trial}}(\mathcal{T}) = T_{\text{audio}} - N_{\text{ref}}(\mathcal{T}) \quad (3)$$

where T_{audio} is the total audio duration in seconds and a rate of one trial per second is assumed.

Varying θ produces a family of $(P_{\text{Miss}}, P_{\text{FA}})$ values that can be visualized as a detection error tradeoff (DET) curve [6] characterizing system performance over a range of operating points; however, for system development it is convenient to have a single operating point at which system performance is optimized. For the STD 2006 evaluation, NIST defined a metric, "term-weighted value," (TWV):

$$\text{TWV}(\theta) = 1 - \frac{1}{|\mathcal{Q}|}\sum_{\mathcal{T}\in\mathcal{Q}}\Big(P_{\text{Miss}}(\mathcal{T},\theta) + \beta P_{\text{FA}}(\mathcal{T},\theta)\Big) \quad (4)$$

where $|\mathcal{Q}|$ is the length of the query list and $\beta = 999.9$ is a weight that reflects the assumed prior probability of term occurrences and the relative costs of misses and false alarms. TWV lies in the range $(-\infty, 1.0]$, with a score of 1.0 corresponding to perfect performance and a score of 0.0 corresponding to no output.

System performance can then be characterized by the actual term-weighted value (ATWV), which is the term-weighted value achieved by a system at the pre-selected threshold that defines the YES/NO decisions in the postings list, and by the maximum term-weighted value (MTWV), which is the best term-weighted value achievable given a *post-hoc* choice of detection threshold.

3. TRAINING AND DEVELOPMENT DATA

The Cantonese speech corpus collected for the Babel program comprises 212 hours of telephony data divided into a 192-hour training set and a 20-hour development set. The development set is further divided into a 13-hour tuning set for selection of keyword search parameters such as the decision threshold and 7-hour validation set. The training set contains 36 hours of scripted material, including dates, times, numbers, person and place names, and phonetically rich sentences, and 156 hours of conversations between subjects who know each other well (friends and family members). The development set contains only conversational material. Approximately 40–50% of the audio is speech.

One of the challenges of the Babel corpora is that they contain, by careful design, a diverse set of speakers. The population of speakers in the Cantonese corpus is 52% female and 48% male; ranges in age from 16–67 years old, with a median of 35; and represents five different dialects, with 24% speaking the Central Guangdong, 20% the Northern Pearl River Delta, 19% the Southern Pearl River Delta, 19% the Guangxi and Western Guangdong, and 18% the Northern Guangdong dialects. Speakers were recorded from six different environments, including quiet home environments, moving vehicles, the street, and public places, and recordings were made from four different telephone networks and at least 50 different handset types.

A second challenge represented by the Babel corpora is sparse language model (LM) training data. There is no separate LM corpus: only the acoustic transcripts are provided. For the Babel Cantonese training set, this amounts to 106K utterances containing a total of 992K word tokens. While participants are allowed to collect additional text data from the web, the systems described here use only the acoustic transcripts.

A lexicon is provided that covers the training and development corpora; it contains 25.7K word types and a total of 29.1K pronunciation variants, for an average of 1.13 pronunciations per word. The lexicon uses a variant of the X-SAMPA phone set, and includes syllabification and annotation with seven different tones. The lexicon does not cover partial words and mispronounced words, so lexicons used for training are somewhat larger.

4. SYSTEM ARCHITECTURE

Our approach to keyword search relies on the combination of keyword search results from multiple indexes [7], where the indexes are produced by a diverse set of speech recognition systems. We achieve system diversity in two ways: we use speech recognition systems constructed by multiple, independent groups, which is a strategy that has worked well for speech recognition [8]; and we use two different types of acoustic model, the standard Gaussian mixture model (GMM) and a deep belief network (DBN) acoustic model. Specifically, we combine search results from indexes produced by four different speech recognition systems: (1) a GMM-based system from IBM (BSRS), (2) a DBN-based system from IBM (DBN), (3) a GMM-based system from Cambridge University (CUED), and (4) a GMM-based system from RWTH Aachen (RWTH). Audio indexing and keyword search are performed using weighted finite state transducer (WFST) methods, and postings lists from the four systems are combined using methods adopted from information retrieval meta-search to produce the final system output.

5. ASR SYSTEMS FOR AUDIO INDEXING

5.1. BSRS system

The IBM BSRS (bootstrap and restructuring [9]) system is a GMM-based speech recognition system that uses four decoding passes with a series of increasingly refined models and unsupervised adaptation after each decoding pass. The four decoding passes are (1) speaker-independent (SI), (2) vocal tract length normalized

(VTLN), (3) speaker-adaptively trained (SAT), and (4) discriminatively trained (DT). Training uses the IBM Attila speech recognition toolkit and follows the recipe described in [10], except that the SAT model uses the bootstrap and restructuring procedure [9] to produce more reliable models from relatively small training corpora. The DT model is based on the SAT model, and includes feature-space and model-space discriminative training using the boosted maximum mutual information criterion. The first three decoding passes produce confusion networks, which are used for confidence-weighted adaptation. The final DT model uses 5000 quinphone context-dependent states and 238K Gaussian mixture components. The feature processing pipeline computes 13-dimensional PLP features with speaker-based mean and variance normalization, splices 9 frames of features and projects to a 40-dimensional feature space using linear discriminant analysis (LDA), and further diagonalizes the class-conditional distributions using a global, semi-tied covariance (STC) transform. Speaker-adaptive training is based on constrained MLLR (CMLLR) [11] transforms, and the final recognition pass uses multiple MLLR [12] transforms. The language model is a back-off trigram model with modified Kneser-Ney smoothing trained on 96K utterances containing a total of 991K words.

5.2. DBN system

The IBM DBN system uses a deep belief network acoustic model and the speaker-adaptive feature processing described above. The DBN takes nine frames of 40-dimensional VTLN PLP+LDA+STC+CMLLR features as input, contains five hidden layers each comprising 2048 logistic units, and has a softmax output with 3000 quinphone context dependent state targets. The DBN training procedure is divided into three steps: (1) discriminative pretraining, (2) training with the cross-entropy criterion, and (3) training with the state-level minimum Bayes risk (sMBR) criterion. In the discriminative pretraining phase, each weight layer in turn is trained using the cross-entropy criterion and only one pass over the data, with the weight layers below being frozen. In the cross-entropy training phase, the weights for the softmax layer are randomly initialized, and then the entire network is trained using a stochastic gradient descent procedure that monitors performance on a held-out set to determine when to reduce the step size. In the sMBR training phase, a distributed implementation [13] of Hessian-free optimization [14] is used to train the network, with progress monitored using the same held-out set as in the cross-entropy training. The lattices from the DBN system are rescored using an interpolated LM combining a neural network language model (NNLM) [15] and a Model M language model [16]. The NNLM is a word-based 4-gram model that uses a 30-d embedding space, has 100 hidden units, and predicts all words in the vocabulary. The Model M LM is an exponential, class-based trigram model that uses 150 automatically generated word classes. More details on the BSRS and DBN acoustic models, and the Model M and NN language models may be found in [17].

5.3. CUED system

The CUED system is a GMM-based speech recognition system that uses two decoding passes: an initial SI pass followed by a second decoding pass with a discriminatively trained SAT model. Training and decoding use the HTK V3.4.1 toolkit with internal extensions produced both prior to and for the Babel program. Acoustic model training used a 162-hour subset of the training data. The second-pass acoustic model uses word boundary and tone dependent phone state-tied cross-word triphone models trained with CMLLR SAT and feature- and model-space discriminative training using the minimum phone error criterion. It contains 6000 states with an average of 16 Gaussian components per state. The feature vectors are 68-dimensional, comprising 52 static, delta, delta-delta, and triple-delta PLP features projected to 39 dimensions with HLDA; pitch with delta and delta-deltas; and 26 bottleneck MLP features. The PLP features are normalized per speaker to have zero mean and unit variance. The MLP features are computed using a network that takes 9 frames of static, delta, delta-delta, and triple-delta PLP features as input, contains two hidden layers of 2000 logistic units each, a 26-unit bottleneck layer, and a softmax output layer with 39 monophone targets. For efficiency the MLP features are incorporated in the same fashion as [18]. Supervision for both global CMLLR and subsequent global MLLR adaptation is based on the initial SI decoding. The language model is a trigram with a vocabulary size of 24K unique words.

5.4. RWTH system

The RWTH system is a GMM-based speech recognition system that uses two decoding passes: an initial SI pass followed by a second decoding pass with a discriminatively trained SAT model. Training was done on the conversational portion of the training set. The acoustic model uses generalized triphone states based on a CART tree that can tie states from different phonemes to a shared emission model. Overall 4501 Gaussian mixture models with a globally tied covariance matrix are used, leading to around 1 million mixture densities that are trained using the minimum phone error criterion. The feature vectors are based on 15 mean- and variance-normalized MFCC features, a voicedness feature, a smoothed tone feature, and MLP features from a hierarchical bottleneck MLP architecture. Separate LDA projections of 9 successive frames are applied for the MLP and the MFCC+voicedness+tone feature streams, before concatenating them to a joint feature stream. The LDA over the MFCC+voicedness+tone features yields 45 dimensions, and the LDA over the MLP features yields 70 dimensions; thus, the final feature stream has 115 dimensions. The audio is preprocessed using a silence normalization method to harmonize the percentage of silence across different segments. A 4-gram Kneser-Ney LM is trained on the transcription of the acoustic training data. To prevent overfitting, the discounting parameters are automatically optimized for the development set [19].

6. WFST KEYWORD SEARCH

Our keyword search toolkit is a two-pass implementation [20] of weighted finite state transducer audio indexing and search [21]. Word lattices from a speech recognition system are processed to generate two indexes: a lexical index in which the alphabet of the WFST is the speech recognition lexicon, and a phonetic index in which the WFST alphabet is the speech recognition phone set.

During search, the query list is split into in-vocabulary (IV) and out-of-vocabulary (OOV) queries, with a query being classified as OOV if any of its constituent words are not in the speech recognition lexicon. For IV queries, each query is converted into a lexical finite-state acceptor and is then composed with the lexical index. For OOV queries, each query is converted to a phonetic finite-state acceptor and is then composed with the phonetic index. Because Cantonese is an ideographic language, the pronunciations for OOV queries are generated using a lookup table that provides pronunciations for each Cantonese character. If an OOV character is encountered in a query, that query is skipped.

In previous work [22, 23, 7] it was found that keyword search performance could be improved by performing query expansion using a model of phonetic confusability. However, to date in our work on the Babel Cantonese task, we have found that this query expansion degrades performance; therefore, no query expansion is used in this work.

7. SCORE NORMALIZATION AND SYSTEM COMBINATION FOR KEYWORD SEARCH

Given postings lists from all four speech recognition systems, we normalize the detection scores and combine the results from all systems to generate a final output [7]. The specific method used in this work, which is described in more detail below, is to (1) apply sum-to-one normalization to each postings list, (2) combine the results using MTWV-weighted CombMNZ fusion, and finally (3) apply sum-to-one normalization to the fused postings list to produce the final output. Experiments that led to this particular strategy are described in detail in [24].

Sum-to-one normalization [25] is applied separately to each term, $\mathcal{T}$, in a postings list. It the list contains $N_{\text{hyp}}(\mathcal{T})$ occurrences of term $\mathcal{T}$ with detection scores $s(i,\mathcal{T}), i = 1 \ldots N_{\text{hyp}}(\mathcal{T})$, then sum-to-one normalization computes new scores

$$\hat{s}(i,\mathcal{T}) = s(i,\mathcal{T}) / \sum_{j=1}^{N_{\text{hyp}}(\mathcal{T})} s(j,\mathcal{T}) \quad (5)$$

In the special case of a single occurrence, the normalized score is defined to be 1.0. The first application of sum-to-one normalization to system-specific postings lists makes the detection scores from different systems comparable to one another. Although the raw scores from different systems should already be comparable because they all are expected counts from indexes generated using the same procedures, in practice we obtain better performance with initial score normalization. The second application of sum-to-one normalization optimizes the term-weighted value for queries with different frequencies of occurrence. Under the TWV metric, the cost of a false alarm is nearly constant, while the cost of a miss is inversely proportional to query frequency. Therefore, it makes sense to assign higher scores to rarer queries [26]. Sum-to-one normalization does this, using the sum of the query detection scores as a proxy for query frequency.

System combination is performed using an MTWV-weighted version [24] of the CombMNZ method [27]. First, detection scores in each postings list are weighted in proportion to the list's MTWV score on the tuning portion of the development set. Then, for each postings list, all term occurrences that overlap in time are fused into a "meta-occurrence", with the start and end times taken from the highest-scoring term occurrence and the score of the meta-occurrence being the sum of the scores of the constituent occurrences. Finally, the results across multiple postings lists are produced by combining temporally overlapping meta-occurrences from the different lists, with the start and end times taken from the highest-scoring meta-occurrence, and the score of the meta-occurrence being the sum of the meta-occurrence scores, weighted by the number of lists that contain the meta-occurrence. The detection threshold is optimized on the tuning portion of the development set, and then ATWV is measured on the validation portion of the development set.

8. RELATION TO PREVIOUS WORK

To our knowledge, combining posting lists from diverse keyword search systems to achieve better results was first done on a noisy Levantine Arabic task as part of the DARPA RATS program [7]. The work presented here shows that this combination strategy is useful on a task with very different challenges and at a very different operating point for keyword search. In the RATS task, the main challenge is severe noise and channel distortion, while in the Babel task the main challenges are speaker variability and severely limited LM training data. Also, in the RATS task, the target false alarm rate is much higher than in the Babel task. Compared to previous work on the STD 2006 evaluation [28, 26], in this work the speech recognition systems make much heavier use of neural networks for feature extraction and acoustic modeling, the keyword search system is based on WFSTs, and the keyword search system combines multiple postings lists.

model	% CER	lattice density	tune MTWV	val ATWV	val MTWV
BSRS	53.0	1200	0.435	0.445	0.446
DBN	48.9	1220	0.487	0.483	0.483
CUED	52.9	4120	0.467	0.453	0.458
RWTH	52.7	2100	0.470	0.465	0.473
all	—	—	0.535	0.517	0.524

Table 1. Performance of the four individual speech recognition systems, and the combination of all four systems, measured in terms of character error rate on the development data, lattice density in arcs per second of audio, and keyword search performance on the tuning and validation portions of the development data.

9. SYSTEM PERFORMANCE

The performance of the four individual speech recognition systems and their combination is summarized in Table 1. All four individual systems have ATWVs on the validation set that are within 8% of one another and character error rates on the full development set that are within 8% of one another. At the same time, the four systems use very different training procedures, acoustic features, and acoustic models. The result is that the combined system's ATWV on the validation set is 7% better than the best single system's. In addition to being an effective means for improving keyword search performance, combination of postings lists is easy to implement and computationally inexpensive.

10. ACKNOWLEDGMENTS

We are grateful to Janice Kim of IBM Research for providing software support for the keyword search toolkit and to Roger Hsiao of BBN, and the Babelon team, for sharing their division of the babel101b-v0.4c development data into keyword search tuning and validation sets. This effort uses the IARPA Babel Program Cantonese language collection release babel101b-v0.4c. 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. 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.

11. REFERENCES

[1] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," in *Proc. ICASSP*, 1990, pp. 129–132.

[2] "NIST spoken term dection portal," `http://www.itl.nist.gov/iad/mig/tests/std/`.

[3] "The 2006 spoken term detection evaluation plan," `http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf`.

[4] J. G. Fiscus, J. G. Ajot, J. Garofalo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proc. SIGIR Workshop on Searching Spontaneous Conversational Speech*, 2007, pp. 51–57.

[5] "IARPA broad agency announcement IARPA-BAA-11-02," 2011, `https://www.fbo.gov/utils/view?id=ba991564e4d781d75fd7ed54c9933599`.

[6] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eurospeech*, 1997, pp. 1895–1898.

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

[8] M. J. F. Gales, D. Y. Kim, P. C. Woodland, H. Y. Chan, D. Mrva, R. Sinha, and S. E. Tranter, "Progress in the CU-HTK Broadcast News transcription system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1513–1525, 2006.

[9] 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 Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2252–2264, 2012.

[10] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. IEEE Spoken Language Technology Workshop*, 2010, pp. 85–90.

[11] M. J. F. Gales, "Maximum-likelihood linear transforms for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[12] C. J. Leggetter and P. C. Woodland, "Speaker adaptation of continuous density HMMs using multivariate linear regression," in *Proc. International Conference on Spoken Language Processing (ICSLP)*, 1994.

[13] 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.

[14] J. Martens, "Deep learning via Hessian-free optimization," in *Proc. Intl. Conf. on Machine Learning (ICML)*, 2010.

[15] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Proc. Neural Information Processing Systems (NIPS)*, 2000.

[16] S. F. Chen, "Shrinking exponential language models," in *Proc. HLT-NAACL*, 2009, pp. 468–476.

[17] J. Cui, X. Cui, J. Mamou, B. Kingsbury, B. Ramabhadran, L. Mangu, M. Picheny, A. Sethy, and J. Kim, "Developing speech recognition systems for corpus indexing under the IARPA Babel program," in *Proc. ICASSP*, 2013. To appear.

[18] J. Park, F. Diehl, M. J. F. Gales, M. Tomalin, and P. C. Woodland, "The efficient incorporation of MLP features into automatic speech recognition systems," *Computer Speech and Language*, vol. 25, pp. 519–534, 2010.

[19] M. Sundermayer, R. Schlüter, and H. Ney, "On the estimation of discount parameters for language model smoothing," in *Proc. Interspeech*, 2011, pp. 1433–1436.

[20] D. Can, E. Cooper, A. Sethy, C. White, B. Ramabhadran, and M. Saraçlar, "Effect of pronunciations on OOV queries in spoken term detection," in *Proc. ICASSP*, 2009, pp. 3957–3960.

[21] C. Allauzen, M. Mohri, and M. Saraçlar, "General indexation of weighted automata — Application to spoken utterance retrieval," in *Proc. HLT-NAACL*, 2004.

[22] U. V. Chaudhari and M. Picheny, "Matching criteria for vocabulary-independent search," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1633–1643, 2012.

[23] J. Mamou and B. Ramabhadran, "Phonetic query expansion for spoken document retrieval," in *Proc. Interspeech*, 2008, pp. 2106–2109.

[24] J. Mamou, J. Cui, X. Cui, M. J. F. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlüter, A. Sethy, and P. C. Woodland, "System combination and score normalization for spoken term detection," in *Proc. ICASSP*, 2013. To appear.

[25] M. Montague and J. A. Aslam, "Relevance score normalization for metasearch," in *Proc. ACM International Conference on Information and Knowledge Management*, 2001, pp. 427–433.

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

[27] J. A. Shaw and E. A. Fox, "Combination of multiple searches," in *Proc. 2nd Text Retrieval Conference (TREC-2)*, 1994, pp. 243–252.

[28] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. SIGIR*, 2007, pp. 615–622.