CALIBRATION AND MULTIPLE SYSTEM FUSION FOR SPOKEN TERM DETECTION USING LINEAR LOGISTIC REGRESSION

J. van Hout[1], *L. Ferrer*[1], *D. Vergyri*[1], *N. Scheffer*[1], *Y. Lei*[1], *V. Mitra*[1], *S. Wegmann*[2]

[1]SRI International, Menlo Park, CA
[2]ICSI, Berkeley, CA

ABSTRACT

State-of-the-art calibration and fusion approaches for spoken term detection (STD) systems currently rely on a multi-pass approach where the scores are calibrated, then fused, and finally re-calibrated to obtain a single decision threshold across keywords. While the above techniques are theoretically correct, they rely on meta-parameter tuning and are prone to over-fitting. This study presents an efficient and effective score calibration technique for keyword detection that is based on the logistic regression calibration approach commonly used in forensic speaker identification. The technique applies seamlessly to both single systems and to system fusion, and enables optimization for specific keyword detection evaluation functions. We run experiments on a Vietnamese STD task, comparing the technique with more empirical calibration and fusion schemes and demonstrate that we can achieve comparable or better performance in terms of the NIST ATWV metric with a more elegant solution.

Index Terms— spoken term detection, score calibration, score normalization, system fusion

1. INTRODUCTION

A pressing need exists for intelligent information retrieval (IR) from the rapidly increasing amounts of audio data being created and broadcasted daily from various sources. Spoken term detection (STD) aims to locate a specified *term*, defined as a sequence of one of more words, rapidly and accurately in large, potentially heterogeneous audio archives, and can be used as a first step to more sophisticated IR technologies.

Most state-of-the-art STD systems involve two major components: indexing and search. The indexing step processes the audio data by an automatic speech recognition (ASR) system which creates a searchable text index with scores. The search step, given an input query term, decides which instances to return to the user. Because the queries may contain out-of-vocabulary (OOV) terms that would be absent in a word-based index, or query terms may be missing in the ASR output due to errors, most STD systems utilize indices that combine word and phonetic information, and thus enable searching to match pronunciation patterns in the index [1]. Since STD is formulated as a detection task, with a metric that considers both misses and false alarms, the system must decide whether to return a hypothesized instance or not by applying a threshold on the scores, which are typically word posteriors provided by the ASR system. Nevertheless, these scores are often not a good indicator of word correctness (confidence). Prior work focused on computing word confidence by using the word posterior along with other features [2], or aimed to apply a keyword specific threshold [3].

In this work, inspired by speaker identification and forensics research, we propose a principled approach for word posterior calibration that follows the Bayesian decision theory and can handle either an individual system, or multiple systems by fusing the scores into a single calibrated score, as well as any available side information (e.g., noise levels, speaker information, etc). The approach optimizes system performance for a desired operating point.

2. TASK DESCRIPTION

The National Institute of Standards and Technology (NIST) created the STD evaluation [4] initiative to provide a benchmark testbed for this task. The IARPA Babel program recently supported a follow-up open evaluation with a focus on the rapid development on a surprise language with limited data. The evaluation plan is described in [5]. This work focused on the FullLP + BaseLR + NTAR condition: the system development used only data from the NIST release IARPA-babel107b-v0.7, and the system did not reprocess the test audio after the query keywords were provided. The system performance was evaluated on the provided query terms, using the Actual Term-Weighted Value (ATWV) metric, defined in [5]. This metric is given by 1 minus the following risk function:

$$R = \frac{1}{K}\sum_{k=1}^{K} P_{\text{miss}}(k) + \beta P_{\text{fa}}(k) \tag{1}$$

where k runs over all keywords with at least one reference occurrence, $\beta = 999.9$, $P_{\text{miss}} = N_{\text{miss}}(k)/N_{\text{true}}(k)$, and $P_{\text{fa}} = N_{\text{fa}}(k)/N_{\text{NT}}(k)$, with $N_{\text{true}}(k)$ denoting the number of reference occurrences of keyword k, $N_{\text{miss}}(k)$ the number of missed detections for k, $N_{\text{fa}}(k)$ the number of false detections for k, and $N_{\text{NT}}(k) = T_{\text{speech}} - N_{\text{true}}(k)$. T_{speech} is the total amount of evaluated speech in the test data.

3. STD SCORE CALIBRATION AND FUSION

This section describes the previous and proposed approaches for calibrating and fusing scores generated by STD systems with the goal of optimizing the ATWV metric.

3.1. Prior work and baseline systems

Since the ATWV metric was introduced in the NIST STD06 evaluation, various approaches have been designed for selecting the threshold that maximizes ATWV. In [3], the authors explain that the optimal keyword-specific threshold (KST) on the posteriors according to Bayesian decision theory is given by

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0014. 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.

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

$$T_{\text{post}}(k) = \frac{\beta N_{\text{true}}(k)}{N_{\text{NT}}(k) + \beta N_{\text{true}}(k)} \quad (2)$$

The value of $N_{\text{true}}(k)$ is unknown during testing and needs to be estimated. In [3], $N_{\text{true}}(k)$ is estimated as $\hat{N}_{\text{true}}(k) = C\mathbf{E}[\text{count}(k)]$, where the expected count is obtained as the sum of the posteriors for all hits of keyword k, and the factor C is estimated on the training data and accounts for the fact that some keyword detections are missing due to lattice pruning. In our experiments, we computed this factor for each system as the ratio of the average $N_{\text{true}}(k)$ and the average $\mathbf{E}[\text{count}(k)]$ over all keywords in a held-out set of 2000 keywords on the training data. This approach will be referred to as KST$_{trn}$. For comparison, we report results using the optimal factor computed on our test data as KST$_{max}$, which provides an upper bound on the performance of the above thresholding technique.

Besides obtaining an accurate estimate of N_{true}, the second important assumption made in the KST approach is that the posteriors for the hits are well calibrated (that is, are proper posteriors). In [6] the authors introduce a technique called "p_{corr} mapping" that learns the true posteriors of the training scores by computing their accuracy over consecutive bins. The piecewise-constant function that computes p_{corr} from the score is smoothed by linear interpolation before being applied to test data. This approach is prone to over-fitting and requires tuning of the smoothing parameters. For this work we implemented an approach introduced in [7] that uses logistic regression to learn a linear transformation on the logit of the score. The resulting transformed score approximates the log-likelihood ratio (LLR) of the score given the two classes (correct and incorrect hit). The class priors found in training are used to transform this LLR into the required posterior. This approach will be referred to as llr$_{cal}$.

State-of-the-art STD systems combine the hit lists originating from multiple ASR systems in a process called system fusion. In [6], the authors propose an approach for aligning the hits and combining the scores into a fused score in a linear fashion in the posterior domain by learning weights for different systems in a way that maximizes MTWV. Although they obtained good results on the training set, this technique did not generalize well on test data and obtained similar performance as rule-based fusion techniques, such as averaging the posteriors of the systems with equal weights, which we refer to as fus$_{avg}$. In this technique, missing posteriors in the aligned hits are assigned a value of 0. As in the case of score calibration, logistic regression has been shown very effective for fusing hit lists from various STD systems, as described in [7] and [8]. This approach is our second baseline fusion approach, and will be referred to as fus$_{llr}$. All of the fusion experiments described in this paper use the same procedure to align hits from individual systems, which will be described in section 3.2.1.

3.2. Proposed Approach

In our proposed approach, the hits generated by the different STD systems are combined into a fused set of hits using a modified linear logistic regression approach. When a single system is put through this process, the output is a set of hits with 'calibrated' scores.

3.2.1. Aligning Hits from Different Systems

The hits from different STD systems are aligned with each other using the algorithm described in [7] and [8]. Given a certain utterance and keyword k, a floating window of 0.8 seconds is shifted across the waveform. For a certain position of the window, all hits of keyword k from all systems that fall within that window are collected, keeping the hit with the highest score for each system. Each window position in which at least one system had a hit is considered a fused hit and is labeled as either a correct hit, $y = +1$, or a false alarm, $y = -1$, with the same procedure used during scoring where a tolerance of 1 second is used to align a certain hit with the reference.

For each hit, a feature vector x of dimension 2N is created where N is the number of individual systems being combined. The first N values in this vector correspond to the logit function of the posteriors generated by the individual systems for the hit. If a system does not produce a hit within the window, a value of M=0 is assigned. The second set of N values correspond to indicator variables that are set to one if a system does not generate a hit within the window.

3.2.2. Making Optimal Decisions

Let us first assume we already have a score for each hit computed as some function of the feature vector x described above. Given these scores we will make the final decision of converting a hit into an actual keyword detection by thresholding its score with keyword-dependent thresholds. The goal is to select the thresholds to minimize the risk given in Eq. (1). This risk is defined by assuming that a detection can be generated for each keyword every second. That is, that if the threshold is set low enough, no misses would occur. On the other hand, our fused hits are defined only over regions in which at least one of the individual systems had a hit. Taking this into account, assuming a single hit for each reference keyword is present in our samples, and discarding a term corresponding to the unrecoverable misses (keywords that were not found by any individual system) and the K factor which do not affect the optimal thresholds, we can rewrite the risk as

$$\tilde{R} = \alpha \sum_{k=1}^{K} p(k)[p_c(k)\tilde{P}_{\text{miss}}(k) + (1 - p_c(k))\tilde{P}_{\text{fa}}(k)] \quad (3)$$

where $\tilde{P}_{\text{miss}}(k)$ and $\tilde{P}_{\text{fa}}(k)$ are the probability of miss and false alarm computed on the fused hits directly. Here we have defined $p(k)$, an effective prior for each keyword, and $p_c(k)$ an effective prior for correct hit, which allow us to see the risk function as a probability of error. The constant α takes whatever value is needed to make $p(k)$ and $p_c(k)$ probability distributions. In order for Equations (1) and (3) to be equivalent the $p(k)$ and $p_c(k)$ need to satisfy:

$$\alpha p(k)p_c(k) = \frac{N_{+1}(k)}{N_{\text{true}}(k)}, \qquad \alpha p(k)(1 - p_c(k)) = \beta \frac{N_{-1}(k)}{N_{\text{NT}}(k)} \quad (4)$$

for all k from 1 to K, where $N_{+1}(k)$ and $N_{-1}(k)$ are the total number of correct and incorrect hits, respectively. As we will see, we do need not explicitly solve these equations.

Because we plan to use different thresholds for each keyword, we can optimize Eq. (3) by minimizing each term in the sum over k independently. Bayesian decision theory indicates that in order to minimize this risk, the system should decide the hit is a detection if and only if:

$$\text{LLR} = \log \frac{p(x|y = +1, k)}{p(x|y = -1, k)} > -\text{logit}(p_c(k)) = T_{\text{LLR}}(k) \quad (5)$$

where LLR (log-likelihood ratio) is the ratio between the likelihood of the features assuming that the hit is a correct hit and that the hit is an incorrect hit. The optimal threshold can be computed using Equations (4) as:

$$T_{\text{LLR}}(k) = \log \left(\beta \frac{N_{-1}(k)}{N_{+1}(k)} \frac{N_{\text{true}}(k)}{N_{\text{NT}}(k)} \right) \quad (6)$$

Except for β, all values in this equation are unknown during testing. Moreover, test keywords are not known during development. Hence, the first ratio, $N_{-1}(k)/N_{+1}(k)$, is replaced by the corresponding ratio over all keywords in the training data and the $N_{\text{true}}(k)$ is estimated by the expected count as described in section 3.1. For this, a preprocessing step of standard logistic regression is done to obtain the posteriors used to compute the expected count. The scale factor for the $N_{\text{true}}(k)$ estimation does not need to be explicitly computed in this approach since, in the log scale it (approximately) corresponds to a bias that is automatically and optimally compensated for by the global bias in the model given by Eq. (7).

The threshold in Eq. (6) can be easily shown to be equivalent to the one defined in Eq. (2) which operates in the domain of the posteriors instead of the LLRs. In the next section we will explain how we estimate the LLRs for the hits given the vector of features x.

3.2.3. *Modified Logistic Regression for TWV Optimization*

Linear logistic regression is a very common approach for converting a vector of features into likelihood ratios. The standard logistic regression assumes that the posteriors for the two classes are given by $P(y|x) = \sigma(y(ax + b))$, where $\sigma(z)$ is the logistic function $1/(1 + e^{-z})$, x is the input feature vector for the sample (a hit in our case), and y is the class label ($+1$ or -1). Parameters a and b are obtained by maximizing the log-likelihood of the training data given by $\sum_i \log P(y_i|x_i)$. It can be easily shown that, given this form for the posterior, the LLR is simply given by $ax + b - \text{logit}(p_c)$ where p_c is, as before, the prior probability of a positive sample (a correct hit, in our case).

This standard form is not ideal for our purposes because it optimizes the transformation given the priors found in the training data. As we saw in the previous section, the risk function we are trying to minimize can be seen as affecting the priors of the classes and the keywords to some synthetic values given by Equation (4) which are keyword dependent. We modify the logistic regression approach to take into account these priors and, hence, optimize the desired risk function given by Equation (3). We assume the following form for the posterior for each hit:

$$P(y_i|x_i, k) = p_{i,k} = \sigma(y_i(ax_i + b + \text{logit}(p_c(k)))) \tag{7}$$

where, $\sigma(z)$ is the logistic function, k is the keyword, x_i is the feature vector for hit i, and y_i is its class label. The LLR needed in Eq. (5) is now given by $ax_i + b$ and the exponent in (7), $ax_i + b + \text{logit}(p_c(k))$, can simply be compared to 0 to make the final decision. The extra term in the definition of the posterior ensures that a single set of a and b can be found that is optimal over all keywords. In practice, we also learn a weight for the $\text{logit}(p_c(k))$ term along with a and b. This weight is very close to 1 but not identical, indicating that the model might be compensating for some inaccuracy in our assumptions. This issue will be investigated in future work.

To obtain the optimal values for the parameters we maximize the likelihood of the training data, though the expression for the likelihood is modified to take into account for the fact that the priors in the training data do not necessarily coincide with the synthetic priors corresponding to the probability of error that we wish to optimize, given by Eq. (3). The modified likelihood can be shown to be:

$$L = \sum_k \left[\frac{\sum_{i|y_i=+1,k} \log p_{i,k}}{N_{\text{true}}(k)} + \frac{\beta \sum_{i|y_i=-1,k} \log p_{i,k}}{N_{\text{NT}}(k)} \right] \tag{8}$$

With this objective function the optimization is done for the effective priors rather than for the ones found in the data. A standard numerical optimization tools (L-BFGS) is used to optimize this objective function. This technique will be referred to as TWV_{cal} when applied to calibrate a single system, and as TWV_{fusion} when applied to fuse multiple systems.

4. EXPERIMENTAL SETUP

4.1. Data

For this work we used the Vietnamese language package provided by NIST for the 2013 OpenKWS evaluation. The training data included approximately 100 hours, and the development set, on which we report results, was approximately 10 hours. We used the keyword set provided for development purposes, which included 200 query terms. To train the calibration and fusion models, we defined a separate keyword list of 2000 keywords, non-overlapping with the 200 dev keywords, whose distribution of priors in the dev data resembled those of the dev keywords. The official keyword list used for the evaluation included about 4000 terms; but for this work, scoring all the systems on the eval set and eval keyword list was impractical, so we report development set results.

4.2. ASR Systems

For our experiments we used word indices produced by ASR systems from SRI and ICSI. SRI targeted the development of multiple systems and constrained the run-time and memory requirement of each to produce output from all of them, with the goal of combining dissimilar indices and thus gaining by system combination. ICSI targeted the development of a single system that would achieve deep enough indexing to offer optimal single-system performance.

4.2.1. *SRI ASR Systems*

The different systems developed at SRI aimed to produce dissimilar indices by using either different front-end features, different acoustic and language modeling approaches and different engines: open-source Kaldi [9] or the SRI-proprietary DECIPHER software [10], as shown in Table 1.

For the front-end, we used the standard PLP and MFCC features but also explored several novel front-ends. All the front-ends used were augmented with a spline smoothed pitch feature [11] (along its 1st and 2nd derivatives). The MFCCs were also augmented with a 10-dimensional voicing feature vector [12]. The three novel features explored were:

(1) The Normalized Modulation Cepstral Coefficient (NMCC) [13], obtained from tracking the amplitude modulations of the sub-band speech signals in time domain. The produced 52-dimensional vector was reduced to 20 with principal component analysis (PCA) (NMCC20).

(2) The Synchronized Damped Oscillator Cepstral Coefficient (SyDOCC) which models the motion of auditory hair cells as forced damped oscillators. It obtains instantaneous damped oscillator responses from a bank of damped oscillators each tuned to a specific frequency, where the forcing function is a band-limited speech signal, and synchrony is induced by coupling neighboring forcing functions after timealignment.

(3) The NMCCs were used as inputs to train a single hidden layer auto-encoder (AE) using 150 neurons in the hidden layer, with tan-sigmoid activation function and scaled conjugate gradient as the learning algorithm. Once trained, we took the 150 dimensional hidden variables and performed PCA to reduce the dimentionality to 53, ensuring that at least 95% of the information was retained. This final verctor was used as the AE feature.

Using different options of the above front-ends, SRI trained three types of acoustic models (AMs): Gaussian Mixture Models (GMMs) using DECIPHER, and subspace GMMs (SGMMs) and a deep neural network (DNN), using Kaldi. Each training/test file (conversation side) was clustered into 2-3 pseudo speaker clusters using unsupervised agglomerative clustering, to capture possible cases of a speaker change. Mean, variance and vocal tract length (VTL) normalization was applied over the pseudo speaker clusters. For SRIb and SRIe the input features (using 1st, 2nd and 3rd derivatives) were transformed using heteroscedastic linear discriminant analysis (HLDA) to 39-dimensional feature vector. For the rest of

SGMM Kaldi systems LDA+MLLT was used, after concatenating feature vectors from seven frames. We used three-state left-to-right HMMs to model crossword triphones. For the GMMs, we trained discriminative models (applying both fMPE and MPE [14]); and for the SGMMs, we trained discriminatively trained (BMMI) models [15]. The GMM models were speaker adapted using maximum likelihood linear regression (MLLR) and the SGMM models using feature-space MLLR and speaker subspace adaptation.

The language model (LM) used all NIST-provided training data, about 1 million syllables. SRILM was used to train syllable 3-grams (syl-3gr) using Kneser-Ney smoothing. We also used an approach exploring n-gram statistics to extract multiwords (mw) from the training data. In each iteration, top word pairs were merged into mws, after considering the geometric mean of the forward bigram conditional probability between two words and their reverse bigram probability. More details can be found in [16]. We learned 4130 mws which expanded the vocabulary size to 10006, and trained the maximum entropy mw-ngram using SRILM toolkit extension [17].

4.2.2. *ICSI ASR System*

ICSI system [18] used tandem features, pasting cepstral, pitch, voicing and bottleneck (BN) features. An LDA transformation was applied to the cepstral part of the feature, taking as input a context of 7 spliced static MFCC vectors and trained using the context dependent states as targets, reducing dimensionality to 30. The 30-dimensional tandem BN features [19] were obtained using a hierarchical NN. MLLT and fMLLR transforms were applied to the combined feature stream. The first two decoding passes were done with a standard continuous GMM AM with 5k context dependent states and 80k Gaussian components. The third and final decoding pass was done with an SGMM model with 8k states and 50k sub-states derived from 700 Gaussians. More details are found in [18]. A syl-3gr LM was estimated using the training transcripts, applying Kneser-Ney smoothing and interpolationed counts.

Table 1. *Summary of the ASR systems used for combination*

ID	front-end	AM	LM	Adaptation
SRIa	PLP	Kaldi SGMM	syl-3gr	from SRIb
SRIb	PLP	DECIPHER GMM	mw-3gr	from SRIa
SRIc	PLP+NMCC20	DECIPHER GMM	mw-3gr	from SRIc
SRId	Sydocc	Kaldi SGMM	syl-3gr	from SRIa
SRIe	MFCC+voic.	Kaldi SGMM	syl-3gr	from SRIa
SRIf	AE	Kaldi SGMM	syl-3gr	from SRIa
SRIg	MFCC+voic	Kaldi DNN	syl-3gr	n/a
ICSI	MFCC+BN	GMM+SGMM	syl-3gr	ICSI

4.3. Calibration

In Figure 1, we report results on calibrating three of our best performing single systems, both in terms of ATWV and MTWV. We observe that TWV$_{cal}$ obtains similar performance to KST$_{trn}$ and llr$_{cal}$+KST$_{trn}$ on average across all three systems. We observe small, probably insignificant variations in the relative performance of each technique across systems. Apart from llr$_{cal}$+KST$_{max}$ which provides an upper bound on the performance of KST, none of the techniques consistently outperform the others.

4.4. Fusion

In Figure 2, we report results of fusion of the proposed ASR-based STD systems. We study two configurations: the first one combines together all seven SRI's systems, while the second combines all of SRI's and ICSI's systems. Three types of approaches to fusion are compared: (1) average fusion on calibrated individual scores, followed by a second round of calibration (llr$_{cal}$+fus$_{avg}$+llr$_{cal}$); (2) logistic regression-based fusion (fus$_{llr}$); and (3) the proposed joint fusion and calibration scheme (TWV$_{fusion}$). For the TWV_{fusion} technique we use a single way of choosing the keyword-dependent thresholds (Section 3.2). For the other two techniques, which return posteriors, we evaluated three different techniques for selecting a threshold: KST$_{trn}$, KST$_{max}$ and TWV$_{cal}$. For both configurations of ASR systems, it is interesting to note that TWV$_{fusion}$ and fus$_{llr}$+TWV$_{cal}$ obtain similar and competitive performance, and are by far the best fusion techniques in terms of ATWV in the configuration that include ICSI's system. When used as a calibration technique on the fused scores, TWV$_{cal}$ significantly outperforms KST$_{trn}$ in all four cases (two systems configurations, and two fusion strategies), and KST$_{max}$ in three out of four cases. This Table also shows that fus$_{llr}$ is more efficient than fus$_{avg}$, especially with ICSI's system. We believe that this difference in performance could be due to the fact that ICSI's system generates much thicker lattices than SRI's systems, which would create many trials with missing scores. This might be a problem for score averaging because it assumes that a missing score has zero value. In both TWV$_{fusion}$ and fus$_{llr}$ we model a bias for missing scores of each modality, which makes these approaches more robust to fusing heterogeneous systems.

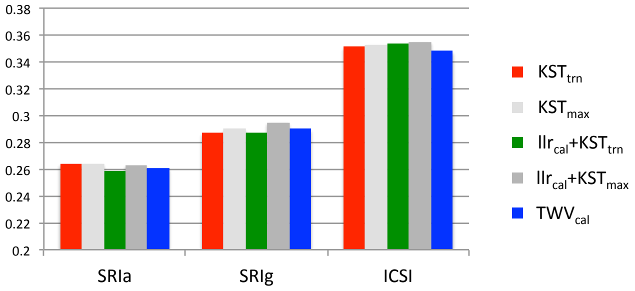

Fig. 1. Calibration results on three of the single systems in terms of ATWV. The techniques using KST$_{max}$ cannot be fairly compared to others and are therefore shown in grey.

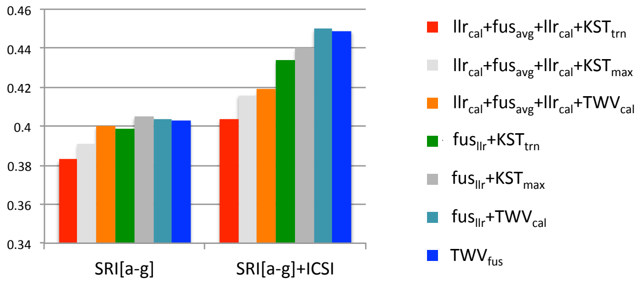

Fig. 2. Fusion results on two different combinations of systems in terms of ATWV. The techniques using KST$_{max}$ cannot be fairly compared to others and are therefore shown in grey.

5. CONCLUSION AND FUTURE WORK

In this paper, we present an approach to jointly calibrate and fuse multiple STD systems in order to maximize the ATWV metric. Our technique can be applied to single-system score calibration or system fusion and provides an approach that is both effective and elegant. We demonstrate that the proposed approach outperforms traditional methods to system fusion and threshold selection such as average fusion, logistic regression fusion and keyword specific thresholding of posteriors. Future work will evaluate in more details the ability of this technique to provide better fusion performance over a wide range of operating points compared to other score normalization schemes.

6. REFERENCES

[1] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *Transactions on Audio, Speech & Language Processing*, pp. 2338–2347, 2011.

[2] Hui Jiang, "Confidence measures for speech recognition: A survey," *Speech communication*, vol. 45, no. 4, pp. 455–470, 2005.

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

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

[5] "Openkws13 keyword search evaluation plan," `http://www.nist.gov/itl/iad/mig/upload/OpenKWS13-evalplan-v4.pdf`.

[6] D. Karakos, R. M. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, L. Nguyen, J. Makhoul, F. Grezl, M. Hannemann, M. Karafiat, I. Szoke, K. Vesely, L. Lamel, and V. Le, "Score normalization and system combination for improved keyword spotting," in *ASRU, to appear*. IEEE, 2013.

[7] M. Akbacak, L. Burget, W. Wang, and J. van Hout, "Rich system combination for keyword spotting in noisy and acoustically heterogeneous audio streams," in *Proc. IEEE ICASSP*, Vancouver, BC, May 2013, pp. 8267–8271.

[8] A. Mandal, J. van Hout, Y-C Tam, V. Mitra, Y. Lei, J. Zheng, D. Vergyri, L. Ferrer, M. Graciarena, A. Kathol, and H. Franco, "Strategies for high accuracy keyword detection in noisy channels," in *Proc. of Interspeech*. ISCA, 2013.

[9] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, and et al., "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[10] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sonmez, and Weng F. et al., "The SRI march 2000 Hub-5 conversational speech transcription system," in *Proc. NIST Speech Transcription Workshop*, 2000.

[11] X. Lei, M. Siu, M.Y. Hwang, M. Ostendorf, and T. Lee, "Improved tone modeling for Mandarin broadcast news speech recognition," in *Interspeech*. ISCA, 2006.

[12] M. Graciarena, H. Franco, J. Zheng, D. Vergyri, and A. Stolcke, "Voicing feature integration in SRI's Decipher LVCSR system," in *ICASSP*. IEEE, 2004, pp. 921–924.

[13] V. Mitra, H. Franco, M. Graciarena, and A. Mandal, "Normalized amplitude modulation features for large vocabulary noise-robust speech recognition," in *Proc. of ICASSP*, Japan, 2012.

[14] J. Zheng and A. Stolcke, "Improved discriminative training using phone lattices," in *Proc. Eurospeech*, Lisbon, sep 2005, pp. 2125–2128.

[15] Daniel Povey, Dimitri Kanevsky, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Karthik Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4057–4060.

[16] X. Lei, W. Wang, and A. Stolcke, "Data-driven lexicon expansion for Mandarin broadcast news and conversation speech recognition," in *Proc of ICASSP*, 2009.

[17] A. Tanel and M. Kurimo, "Efficient estimation of maximum entropy language models with n-gram features: an SRILM extension," in *Proc. of Interspeech*, Chiba, Japan, 2010.

[18] A. Wegmann, S. ad Faria, A. Janin, K. Riedhammer, and N. Morgan, "The Tao of ATWV: Probing the mysteries of keyword search performance," in *to appear in Proc. IEEE Workshop on Speech Recognition and Understanding (ASRU)*, 2013.

[19] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottleneck features for LVCSR of meetings," in *Proc. ICASSP*, 2007, pp. 757–760.