

RESCORING CONFUSION NETWORKS FOR KEYWORD SEARCH

Victor Soto¹ Erica Cooper¹ Lidia Mangu³ Andrew Rosenberg² Julia Hirschberg¹

¹ Columbia University {vsoto, ecooper, julia}@cs.columbia.edu

² Queens College/CUNY andrew@cs.qc.cuny.edu

³ IBM mangu@us.ibm.com

ABSTRACT

We introduce a two-stage cascaded scheme to rescore Confusion Networks (CNs) for Keyword Search in the context of Low-Resource Languages. In the first stage we rescore the CN to improve the error rate of the 1-best hypothesis using a large number of lexical, phonetic, false alarms and structural features. Using a rank learning Support Vector Machine classifier, we obtain WER gains between 0.54% and 2.84% on Cantonese, Tagalog, Turkish, Pashto and Vietnamese. In the second stage we generate keyword hits from the rescored CN and use logistic regression to detect true hits and false alarms. We compare these to hits generated from the unrescored CN and obtain gains between 0.45% and 0.9% on the MTWV metric by using the mentioned features and including acoustic and prosodic features on Tagalog, Turkish and Pashto.

Index Terms— error detection, error correction, confusion networks, posting lists, rescoring, keyword search

1. INTRODUCTION

Spoken term detection systems typically work from transcripts produced by Automatic Speech Recognition (ASR) engines to identify key words and phrases from speech corpora. However, ASR errors degrade the performance of keyword search when terms sought are not in the recognizer’s top-ranked hypothesis. Recent work ([1, 2, 3, 4, 5, 6, 7, 8]) has attempted to improve recognition accuracy for keyword search and other tasks using discriminative post-processing on recognition output by examining additional features beyond those ones used in recognition. Our work is performed in the context of the IARPA Babel research program, which has as its goal the rapid development of speech and keyword search technologies for Low-Resource Languages – languages for which few computational resources are currently available. We describe here a novel two-stage post-processing approach to improve keyword search results. In the first stage, we classify and rerank ASR output in the form of Confusion Networks (CNs) ([9, 10, 11]). In the second stage, we classify and rescore posting list entries using acoustic features. In this paper, we describe the task in Section 2. We discuss related work in Section 3. In Section 4 we describe the CNs we use in our work and

This research was partially supported by ‘la Caixa’ Fellowship Grant for Post-Graduate Studies, Caixa d’Estalvis i Pensions de Barcelona, and 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.

in Sections 5 and 6 we present our two-stage approach to improving spoken keyword search and our results. We conclude in Section 7 and describe future research.

2. TASK, DATA, AND METRICS

The IARPA Babel program [12] focuses on the rapid creation of speech technology for a diverse set of languages with only a small amount of training data. This research addresses the spoken keyword search task defined in the program, which is to identify all the exact matches for some set of query terms, provided as text, in a given corpus of speech.

The data we use consists of both conversational and scripted telephone speech. We currently focus only on the conversational speech, which is comprised of conversations of approximately 10 minutes in length between two speakers who were recorded on separate channels. The data includes a diverse set of speakers in terms of age and dialect, and has an approximately even gender ratio. A variety of recording conditions are represented in the data. We train our classifiers on data from the development set of both the Full and Limited Language Packs (LPs) supplied for the project for each language. The Full Language Pack development set consists of about 40 hours of speech for all languages except for Vietnamese, which has 20 hours, and the Limited Language Pack is a 10-hour subset of the Full LP speech. Orthographic transcriptions and a pronunciation lexicon are also provided with the data. We evaluate on the “evalpart1” evaluation partition of each LP (for which transcriptions are available), which contains about 5 hours of speech in each language. The languages for our current experiments are Tagalog, Turkish, Pashto, Cantonese, and Vietnamese. LLPs for these language include IARPA Babel Program language collections IARPA-babel{101b-v0.4c, 104b-v0.4bY, 105b-v0.4, 106b-v0.2g, 107b-v0.7}.

Results for keyword search are presented in a Posting List (PL), which consists of a list of all hits found for each keyword in the query list. Each hit is labeled with the audio file in which it was found, the start and end time of the segment of audio that contains the keyword, and a score for that hit. Posting lists are evaluated using Term-Weighted Value (TWV)[13], a weighted function of misses and false alarms that penalizes misses more strongly:

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

TWV is computed as a function of θ , a decision threshold for determining whether the score of any posting list entry qualifies it as a hit, and β , the weight for false alarms. Actual term-weighted value (ATWV) is the TWV at a fixed θ . Maximum term-weighted value (MTWV) is the maximum TWV over all possible values of θ .

3. PRIOR WORK

There has been considerable research investigating the rescore of ASR output and the use of CNs to improve speech recognition and downstream Natural Language Processing tasks. Mangu et al. [1] used transformation-based learning and lexical features to improve WER from the two best hypothesis in a CN confusion bin. Similarly, [2] detects errors on broadcast news transcriptions using lexical, syntactic and contextual information. Tur et al. [3] trained conditional random fields using CNs instead of the 1-best transcription to improve accuracy in slot-filling in semantic frames, improving f-score by 6%. Stoyanchev et al. [4] used syntactic and prosodic features to identify mis-recognized words to generate clarification questions in speech-to-speech translation, obtaining a 40% improvement in f-measure over ASR posteriors. Pincus et al. [5] also sought to identify misrecognized words using lexical, positional, prosodic, semantic, and syntactic features, improving f-measure by 3.9% over the ASR posterior baseline. Nakatani et al. [6] used long contextual information from CNs in a two-step error correction procedure. The first step detected errors using a Conditional Random Field classifier trained on n-gram features of the CNs, and the second used Latent Semantic Analysis to include longer-range contextual features. Using long-range context improved WER by 3.64 points. Novotney et al. [7] employed a semi-supervised learning approach to estimate language model probabilities for out-of-training terms from automatically-recognized audio, resulting in 70% of the gain that would be possible by using manual transcriptions of the same data.

Much research has been devoted to building effective keyword search systems in the context of the current IARPA Babel program. Zhang et al. [8] used probability of false alarm (pFA) to normalize keyword search scores to ensure greater consistency across different keywords than just the ASR posterior. Especially in a Low-Resource task, different words will have different amounts of training data, which will affect the consistency of the acoustic and language model scores. pFA normalization was proposed as a more consistent way to compare hits for different keywords. pFA normalization gave large improvements over a baseline with no normalization – 3.8% in percent of miss and 0.11% for percent of false alarms. Another normalization technique that gives large MTWV gains is sum-to-one normalization (STO), where the score of each hit is normalized such that the scores for all hits for a given keyword sum to one [14].

The speech recognizer whose output we used in our experiments was the IBM Speaker-Adapted DNN (SA DNN) system. This uses a deep neural network (DNN) acoustic model with the standard front-end pipeline [15]. The DNN takes 9 frames of 40-dimensional speaker adapted discriminative features as input, contains 5 hidden layers with 1,024 logistic units per layer, and has a final softmax output with 1,500 targets. Training occurs in three phases: first, layer-wise discriminative pre-training using the cross-entropy criterion, second, stochastic gradient descent training using back-propagation and the cross-entropy criterion, and third, distributed Hessian-free training using the state-level minimum Bayes risk criterion [16]. The lexicon is provided with the training data, and the vocabulary contains only words from this data. The language model (LM) is a trigram LM with modified Kneser-Ney smoothing, trained only on the acoustic transcripts. The lattices are produced using a dynamic decoder [17], and are converted to confusion networks.

In our work we present a strategy to improve both the transcription error rate (TER) of the 1-best word CN and the keyword search performance by using lexical, phonetic, probabilistic, acoustic, prosodic and structural features. In all cases, the set of features to use are easy to obtain for Low-Resource Languages.

4. CONFUSION NETS

Confusion Networks (CNs) (Mangu et al., [10]) are a compact representation of ASR output lattices that are designed to facilitate optimizing for word error rate instead of sentence error rate. CNs are created out of lattices by clustering lattice edges into an ordered series of “bins” representing equivalence classes that are sets of alternate word hypotheses. This clustering of edges into bins is done in an heuristic way, since otherwise no efficient solution is known. First, bins are initialized by putting all lattice edges with the same word label and the same start and end times into the same bin. Then, classes containing different words are merged based on time similarity. When a bin has multiple edges with the same word label, these edges are collapsed into a single edge and their posteriors are combined by adding. The resulting CN has a total ordering on the bins that is consistent with the original lattice.

CNs better allow for the minimization of word error rate because one can use dynamic programming to efficiently compute the alignment and edit distance between the reference string and the confusion network. No efficient algorithm for accomplishing this with lattices is known to exist. CNs also represent ASR output in a more compact way than lattices, without sacrificing accuracy. They also lend themselves well to discriminative rescoring strategies such as ours. Mangu et al. [10] found that the correct edge is top-ranked in its bin over 60% of the time, and it is second-ranked over 10% of the time. The correct edge is extremely rarely ranked outside of the top 10. This suggests that there is room for improvement in word error rate if we can reorder the top edges in a bin effectively.

5. RESCORING CNS

In this section we describe the first stage of our CN rescoring process: rescoring the arcs of the word CNs (WCN). Our objective is two-fold: 1) to improve the TER for ASR performance and 2) to obtain better scores to feed into the posting list rescoring stage. We begin by aligning WCN and transcriptions so as to minimize the Levenshtein distance between both. Each arc is then labeled as correct (+) or incorrect (-) and a feature vector is computed for each.

5.1. Feature Extraction and Selection

We extract a variety of features at the arc, bin, segment, and conversation level. Feature sets are described below:

Lexical Features: This set of features encodes information at the arc level without context, eg. the percentile of the word frequency in the transcriptions, whether the arc is labeled as a silence, an epsilon, or a non-speech tag, the number of syllables of the token, and the syllable index of its primary and secondary stress.

Phonetic Features: We use the phone set given in the LLP of each language to incorporate the count of phones of each word in its lexicon entry as well as and four binary features indicating whether the word begins/ends in an unvoiced consonant or glottal stop.

Syntactic Proxies Features: To approximate syntactic information we use Chen’s [18, 19] model M approach. Model M creates a class-based n-gram language model in which each word belongs to a single class and the prediction of each word depends on previous words and classes. The n-gram language model is then reduced using class information such that similar words in the same context belong to the same class. We include the class to which a token belongs in our feature vector.

Probabilistic Features: At the arc level we compute the probability of the arc being correct, the probability of the arc being correct

	Cantonese	Pashto	Turkish	Tagalog
CN	CN Posterior	CN Posterior	CN Posterior	CN Posterior
	P(token + dev)	P(token + dev)	# ph(?)	# ph(6w)
	token length chrs	pFA	P(token recognized dev)	P(token + dev)
	model-M	# arcs	# apps	# ph(D)
	# appearances	model-M	ends glottal stop	# ph(3)
	P(token + tokens in bin)	ends with glottal stop	starts glottal stop	P(token + tokens in bin)
	percentile word freq.	non-speech tag?	model-M	# arcs (bin)
	# ph(kw)	reranked Posterior	non-speech tag?	model-M
	segment duration	P(token + tokens in bin)	P(token + tokens in bin)	non-speech tag?
	isItEnglish?	# apps	percentile word freq.	percentile word freq.

Table 1. Most prominent features for CN reranking (Limited LP) according to QPFS.

given the other tokens in the confusion bins, and the probability of the arc being correct given the set of phones in its pronunciation lexicon entry. We also compute rank-normalized probabilities of false alarm (pFA) (kw, ps) for each pair of word kw and posterior score ps in the corpus, following [8] and global re-ranked posterior scores rPS(ps) as described in [20]. rPS(kw, ps) is computed by first mapping the pair (kw, ps) to its rank r and then mapping back the rank to the average of the posteriors scores with that rank. Re-ranked posterior scores are critical in detection tasks with global thresholds because they expand the posterior score space of a specific keyword to a global set of scores independent of the keyword.

Structural Features: This set of features are extracted directly from the CNs. At the arc level, they include the posterior score, the arc rank and the ratio between the arc rank and the confusion bin size. At the bin level we include the confusion bin size, the bin number at the segment and conversation level, the distance in seconds and bins to the previous and next silence and to the beginning and end of the segment and conversation. We also include the number of prior appearances of the token in the segment and conversation.

Scripted Term Features: Each LLP includes a list of spoken terms that were scripted and recorded for the non-conversational portion of the LLP. These terms are classified into different categories. Among others, some of the categories contain terms of address, digits and numbers, spelled words, money amounts, times, dates, etc.

Metadata Features: Information about the environment and speakers is also available in the LLP. We use the type of recording location (home/office, public space, vehicle, and so on), the gender and age of the speakers, and the dialect of the language spoken during the conversation.

5.2. Feature Selection

We apply feature selection to reduce the size of our training set, to alleviate computational requirements for the Babel program’s evaluation stage, and to improve the performance of our classifiers. In this work we report feature selection results using Quadratic Programming Feature Selection (QPFS) [21]. This technique optimizes the relevance of the selected features to the class labels and minimizes redundancy among the selected feature set while using the Nystrom method for matrix diagonalization to keep the computational complexity below the cubic of the feature space size. Table 1 shows the 10 best-ranked features for each language based on its LLPs the eval-part1 partition (Vietnamese is not included due to lack of space). We find that the CN posterior scores, the probabilistic features, the model-M classes, the percentile of word frequency and the indicators of non-speech, silence and epsilon arcs are the best features over all. For Pashto and Turkish the indicators of glottal stops

	Full LP		Limited LP	
	baseline	reranked	baseline	reranked
Cantonese	55.2	54.23	67.29	65.47
Pashto	55.55	54.37	67.52	66.38
Turkish	55.16	52.82	69.3	66.46
Tagalog	50.37	48.82	64.3	63.64
Vietnamese	61.42	60.86	72.38	71.67

Table 2. TER results for every language in the Base Period using both the Limited and Full language packages.

are also very important. Some other features highly ranked are the number of appearances of a word before the current arc and the ratio of the arc rank divided by the number of arcs in the confusion bin. Metadata features, except for the dialect identifier, are not relevant.

5.3. Experiments and Results

Since our goal in this first stage is to choose the correct arc from a pool of arcs in the confusion bin, we set up the detection problem as a ranking task on the pool of arcs of a single confusion bin, and choose the highest scoring arc as correct. To do this we use Support Vector Machine’s SVM^{rank} , a highly efficient ranking algorithm contained in the SVM^{light} library [22, 23]. SVM^{rank} performs pair-wise classification for each arc in a confusion bin and assigns real-valued scores. Table 2 shows the TER values for each language and both Language Packs. Each subtable shows the baseline TER using the 1-best confusion network, the TER using the 1-best rescored CNs obtained by SVM^{rank} and the TER gain, in that order. We obtain positive gains in every language ranging from 0.56 to 2.34 for the Full LP and 0.71 to 2.84 for the LLP. It is notable that each language seems to behave differently after CN rescoring; Turkish and Vietnamese always report the best and worst gains respectively. While Cantonese shows the most improvement on the LLP, Tagalog shows the opposite behavior. All ten experiments show statistically significant differences from the baselines under the paired t-test, the paired Wilcoxon test, and the signed test for $p < 0.05$.

6. RESCORING POSTING LISTS

In the second stage of our rescoring procedure we again follow a machine learning approach. Since our posting lists are CN-based, PL entries can be matched to the CNs to find the arcs where the keyword hits occur. Using this strategy however has two important problems: 1) out-of-vocabulary words will not be rescored, since only words in the ASR lexicon appear in the CNs and we need to find the specific

	PFA			PMISS			MTWV		
	b	r-CN	r-PL	b	r-CN	r-PL	b	r-CN	r-PL
Pashto	1.1e-4	9e-5	1e-4	.655	.675	.648	23.83%	23.69%	24.73%
Turkish	1.2e-4	7e-5	1e-4	.552	.613	.567	32.83%	31.39%	33.28%
Tagalog	1.1e-4	1.1e-4	9e-5	.550	.561	.565	34.07%	33.34%	34.54%

Table 3. PFA, PMISS and MTWV results for IV keywords on Pashto, Turkish and Tagalog in the Limited LP conditions.

arcs in each confusion bin, and 2) there is no mechanism to recover missed hits.

6.1. Feature Extraction And Selection

For this stage we extract features from the PL entries and their matched confusion bins. Among other features, we include the score and duration of the entry, the original and rescored posteriors from the matched confusion bins aggregated using different functions (mean, standard deviation, geometric mean, product, max and min), and structural CN features from section 5.1 including the number of bins, the number of arcs, the average number of arcs, the average number of epsilon arcs, the number of tokens and the ratio between the number of matched bins and the number of tokens in the keyword term. Rank-normalized PFA and re-ranked posterior scores are also included in the feature set, although now they are only computed for specific set of keywords being examined. Two new set of features are also introduced at this stage:

Acoustic Features: We extract the pitch contour of the confusion bin segment and compute its median, mean, standard deviation, maximum and minimum, the number of unvoiced cycles in the segment and its percentage, the harmonics to noise ratio (dB) and noise to harmonics ratio, and the autocorrelation of the pitch contour. We also extract the pulses and include the number of pulses, the number of periods and their mean and standard deviation, along with the number of voice breaks and their percentage. Finally we include jitter values (local, local in seconds, its relative average perturbation (RAP) and its 5-point period perturbation quotient) and shimmer values (local, local in dB, and its 3, 5, and 11- amplitude perturbation quotient) as computed in Praat. All acoustic features are normalized at the segment level.

Prosodic Features: *Intonational phrase boundaries* and *pitch accents* are detected using the AuToBI tool for prosodic event detection [24]. Due to the lack of prosodic annotation in the Babel corpus, we use cross-language models trained on Standard American English, German, Italian, and Mandarin for phrase boundary detection task [25] and Standard American English, French, German, and Italian for the accent detection task [26].

We use QPFS once more for feature selection. For this stage, the top 10 best-ranked features (not shown here due to limited space) for posting list entry classification as selected by QPFS is homogenous and shows about the same features in the top positions for every language. Mainly the pFA, re-ranked posterior scores and product and minimum of the rescored arcs are highest valued. Rescored posterior scores appear consistently higher in the ranking than their original counterparts, indicating that they are better suited for the classification task. Also consistently high-ranked are the ratio of epsilon arcs divided by the number of total confusion bins, the geometric mean of the rescored and original posteriors and the ratio between the number of tokens in the keyword divided by the number of arcs in the matched confusion bins. From the set of acoustic features, there is a small subset of them appearing consistently in the top 20. These are: number of voice breaks, shimmer (local in dB), mean number of pe-

riods, mean pitch autocorrelation, mean pitch, percentage of voice breaks and local jitter for Pashto; number of voice breaks, mean pitch, harmonics-to-noise ratio, mean number of periods, number of periods and mean pitch for Turkish; and mean pitch, percentage of degree voice breaks, mean pitch autocorrelation, shimmer 3-APQ and standard deviation of the number of periods for Tagalog.

6.2. Experiments and Results

We use Logistic Regression classifiers from the LIBLINEAR library [27] for the classification task. Given the considerable skew of the corpus towards false alarm entries (97% compared to only 3% of true hits) we train our classifiers by weighting the cost parameters inversely to the class distribution, so that the classifier is able to focus on learning true hit examples. Furthermore, model selection is performed by optimizing the F-measure $f_\beta = (1 + \beta)^2 (p \cdot r) / (\beta^2 p + r)$ of the true hits, for $\beta = 1/2$ so as to give recall double weight with respect to precision.

Results are reported in table 3 for Pashto, Turkish and Tagalog. Results for Cantonese and Vietnamese are missing for lack of syllable-to-token indices and PLs, respectively. The three subtables contain best values for probability of false alarm (PFA), probability of miss (PMISS) and Maximum Term-Weighted Value (MTWV) in that order. Each subtable contains results for the baseline CNs (b), the rescored CNs (r-CN) and the two-stage cascaded rescoring (r-PL). In all the experiments reported here, the scores were normalized using sum-to-one, for an MTWV gain of 5–9% with respect to the raw scores. In all three cases r-PL improves the baseline MTWV by a margin between 0.45 and 0.9 points. r-CN never improves the MTWV over the baseline. This is due to SVM^{rank} producing very low scores to predicted false alarms, thus reducing the number of real false alarms but increasing the probability of missing keyword considerably. The r-CN strategy seems to work well for TER, given the reported results, but not for the KWS task using MTWV.

7. CONCLUSIONS & FUTURE WORK

We have presented a two-stage cascaded approach for rescoring spoken keyword search based on rescored CNs. In the first stage CN arcs are rescored to improve TER by detecting the correct arc in the confusion bin. In the second stage, the rescored CNs are used to extract features to predict true hits and false alarms from the posting lists. Both stages showed improvements of TER (0.56-2.84%) and MTWV values (0.45-0.9%) respectively. We used a large number of features for both tasks. For the first task, lexical and syntactic language dependent features proved to be more relevant, while in the second stage probabilistic features like pFA and reranked posterior scores and structural features were more relevant. Furthermore, all features used in this work are relatively easy to obtain for Low-Resource Languages. In future work we plan to extend the rescoring capabilities of our algorithm to OOV words by mapping the confusability transducer output to the selected confusion net arcs and run cross-language experiments testing on unseen languages.

8. REFERENCES

- [1] L. Mangu and M. Padmanabhan, "Error corrective mechanisms for speech recognition," in *Proc. of Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2001, vol. 1, pp. 29–32.
- [2] A. Allauzen, "Error detection in confusion network," in *Proc. of Interspeech*. ISCA, 2007, pp. 1749–1752.
- [3] G. Tur, A. Deoras, and D. Hakkani-Tur, "Semantic Parsing Using Word Confusion Networks With Conditional Random Fields," in *Proc. of Interspeech*. ISCA, 2013, pp. 2579–2583.
- [4] S. Stoyanchev, P. Salletmayr, Y. Jingbo, and J. Hirschberg, "Localized Detection of Speech Recognition Errors," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 25–30.
- [5] E. Pincus, S. Stoyanchev, and J. Hirschberg, "Exploring Features For Localized Detection of Speech Recognition Errors," in *Proc. of the SIGDIAL Conference*. ACL, 2013, pp. 132–136.
- [6] R. Nakatani, T. Takiguchi, and Y. Ariki, "Two-step Correction of Speech Recognition Errors Based on N-gram and Long Contextual Information," in *Proc. of Interspeech*. ISCA, 2013, pp. 4–7.
- [7] S. Novotney, I. Bulyko, R. Schwartz, S. Khudanpur, and O. Kimball, "Semi-Supervised Methods for Improving Keyword Search of Unseen Terms," in *Proc. of Interspeech*. ISCA, 2012, pp. 3–6.
- [8] B. Zhang, R. Schwartz, S. Tsakalidis, L. Nguyen, and S. Matsoukas, "White Listing and Score Normalization for Keyword Spotting of Noisy Speech," in *Proc. of Interspeech*. ISCA, 2012, pp. 1832–1835.
- [9] L. Mangu, E. Brill, and A. Stolcke, "Finding Consensus Among Words: Lattice-Based Word Error Minimization," in *Proc. of Eurospeech*. ISCA, 1999, pp. 495–498.
- [10] L. Mangu, E. Brill, and A. Stolcke, "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks," *Computer Speech & Language*, vol. 14, pp. 495–498, October 2000.
- [11] L. Mangu, H. Soltau, H. Kuo, B. Kingsbury, and G. Saon, "Exploiting Diversity for Spoken Term Detection," in *Proc. of Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8282–8286.
- [12] M. Harper, "IARPA Solicitation IARPA-BAA-11-02," 2011.
- [13] *OpenKWS13 Keyword Search Evaluation Plan*, 2013, <http://www.nist.gov/itl/iad/mig/upload/OpenKWS13-evalplan-v4.pdf>.
- [14] 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. of Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8272–8276.
- [15] H. Soltau, G. Saon, and B. Kingsbury, "The IBM Attila speech recognition toolkit," in *Proc. SLT*, 2010, pp. 97–102.
- [16] 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.
- [17] H. Soltau and G. Saon, "Dynamic Network Decoding Revisited," in *Proc. ASRU*, 2009.
- [18] S. F. Chen, "Performance prediction for exponential language models," in *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. ACM, 2009, NAACL '09, pp. 450–458.
- [19] S. F. Chen and S. M. Chu, "Enhanced word classing for model m," in *Proc. of Interspeech*. ISCA, 2010, pp. 1037–1040.
- [20] D. Karakos, R. Schwartz, and S. et al Tsakalidis, "Score Normalization and System Combination for Improved Keyword Spotting," in *To appear in Workshop on Automatic Speech Recognition & Understanding (ASRU)*. IEEE, 2013.
- [21] I. Rodriguez-Lujan, R. Huerta, C. Elkan, and C. S. Cruz, "Quadratic Programming Feature Selection," *Journal of Machine Learning Research*, vol. 11, pp. 1491–1516, August 2010.
- [22] T. Joachims, "A support vector method for multivariate performance measures," in *Proc. of the 22nd Int'l Conference on Machine Learning*. ACM, 2005, pp. 377–384.
- [23] T. Joachims, "Training linear SVMs in linear time," in *Proc. of the 12th Int'l Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2006, pp. 217–226.
- [24] A. Rosenberg, "AuToBI A Tool for Automatic ToBI annotation," in *Proc. of Interspeech*. ISCA, 2010, pp. 146–149.
- [25] V. Soto, E. Cooper, A. Rosenberg, and J. Hirschberg, "Cross-language Phrase Boundary Detection," in *Proc. of Int'l Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8460–8464.
- [26] A. Rosenberg, E. Cooper, R. Levitan, and J. Hirschberg, "Cross-language Prominence Detection," in *Proc. of 6th Int'l Conference on Speech Prosody*. ISCA, 2012.
- [27] F. Rong-En, C. Kai-Wei, H. Cho-Jui, W. Xiang-Rui, and L. Chih-Jen, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, June 2008.