Low-Resource Open Vocabulary Keyword Search Using Point Process Models

Chunxi Liu,[1] *Aren Jansen,*[1,2] *Guoguo Chen,*[1] *Keith Kintzley,*[1] *Jan Trmal,*[1] *Sanjeev Khudanpur*[1,2]

[1]Center for Language and Speech Processing & Department of Electrical and Computer Engineering,
[2]Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD USA

{chunxi, aren, guoguo, kintzley, yenda, khudanpur}@jhu.edu

Abstract

The point process model (PPM) for keyword search is a whole-word parametric modeling framework based on the timing of phonetic events rather than the evolution of frame-level phonetic likelihoods. Recent progress in PPM training and decoding algorithms has yielded state-of-the-art phonetic search performance in high-resource settings, both in terms of accuracy and computational efficiency. In this paper, we consider PPM application to low-resource settings where the amount of transcribed speech is severely limited and the pronunciation dictionary is incomplete. By using (i) state-of-the-art deep neural network acoustic models to generate phonetic events and (ii) grapheme-to-phoneme conversion to generate pronunciations for out-of-vocabulary (OOV) keywords, we find the PPM system reaches state-of-the-art OOV search performance at a small computational cost. Moreover, due to their complementary methodologies, combining PPM outputs with the LVCSR baseline produces average relative ATWV improvements of 7% and 50% for in-vocabulary and OOV keywords, respectively (16% overall).

Index Terms: keyword search, point process model, OOV keywords, system combination

1. Introduction

The primary goal of the IARPA Babel Program is to develop scalable multi-lingual keyword search (KWS) capabilities with limited access to the typical linguistic resources that state-of-the-art speech recognition technologies strongly rely on. The dominant mode of the program's research thus far has been adapting the high-resource LVCSR-based keyword search systems that were developed for the NIST 2006 STD evaluation to this low-resource setting. However, with the present restricted availability of transcribed speech for language model estimation and highly incomplete pronunciation lexicons producing high keyword OOV rates, the main strengths of LVCSR for search are substantially handicapped. These programmatic constraints thus provide an opening for previous-generation lightweight phonetic search methods to play a continued role.

Originally presented in [1], the point process model (PPM) for keyword search is a whole-word acoustic modeling and search technique. The PPM is founded on the hypothesis that the timing of robustly identifiable phonetic events provides sufficient cues to decode the underlying linguistic message, which in the present case are occurrences of a given keyword. The PPM trades pronunciation-derived hidden Markov modeling of frame-level phonetic likelihoods for inhomogeneous Poisson process rate parameters characterizing the likelihoods of phonetic event arrivals throughout the keyword. Past studies have demonstrated that sparse phonetic event-driven PPMs permit unprecedented speeds in search collection indexing [2] and improved robustness to noise [3]. Moreover, the PPM was demonstrated to outperform competing phonetic fast lattice search methods in both search speed and accuracy [2].

In this paper, we consider the application of PPM-based keyword search technology to the low-resource multilingual setting of the Babel program. To participate in this challenge space, we consider multiple extensions to the basic framework. First, like hidden Markov model (HMM) based lexical models, the PPMs require a frame-level phonetic acoustic model to generate the phonetic event streams. Thus, we evaluate PPM performance in conjunction with a truly state-of-the-art deep neural network (DNN) acoustic model tailored to the present low resource setting. Second, the original PPM framework required keyword training examples to estimate Poisson rate parameters, while the recently proposed MAP estimation technique allows back-off to a dictionary-derived prior [4]. Given the present preponderance of out-of-vocabulary keywords (which are also out-of-training), we evaluate the use of a grapheme-to-phoneme conversion tool to seed dictionary-based PPMs. Finally, to evaluate LVCSR search complementarity, for the first time we consider the system combination potential of our PPM keyword search system.

Incorporating the above PPM extensions, we perform a comprehensive keyword search evaluation on five Babel languages: Haitian, Lao, Zulu, Assamese, and Bengali. Our baseline is the Kaldi LVCSR-based keyword search system developed by the Johns Hopkins University Babel team [5], which is outfitted with the identical DNN acoustic model we use for the PPM. We decompose search performance into in-vocabulary (IV) and out-of-vocabulary keyword sets, comparing OOV performance against a recently proposed state-of-the-art technique called proxy keyword search [6], which derives putative hits from word lattices. For completeness, we begin with a brief review of the point process model for keyword search. In Section 3, we describe the three components of our low-resource PPM recipe. Finally, in Section 4, we describe our experimental setup and present the results of our evaluation.

2. Point Process Models for KWS

The PPM framework for keyword search first transforms input speech acoustic features into a phone posteriorgram representation. Phonetic events are subsequently selected as the local maxima of the smoothed posterior trajectories exceeding a threshold [7], which distills dense frame-level phonetic likelihood estimates into a minimal set of discrete phonetic sequences in time. This collection of events provides the index of the search collection. Given the phonetic pronunciation of

each keyword, a PPM can be constructed entirely based on the phonetic pronunciation provided by a dictionary [4]. The arrival of phonetic events during the course of a given word utterance is modeled as a collection of inhomogeneous Poisson process, one per phone. By modeling each time-varying Poisson rate function as a mixture of Gaussians, we can employ maximum a posteriori (MAP) estimation of the means, variances, and mixture weights. This MAP estimate functions to fold in observed event timing patterns of any available word exemplars present in the training corpus [4]. The PPM also requires a background model for likelihood normalization; here, we assume that outside the keyword of interest, phonetic events are generated by a homogeneous Poisson process governed by a single independent rate parameter for each phone.

For a given keyword w and candidate keyword occurrence time t, we denote the set of events arriving in the interval $(t, t+T]$ by $O_{t,T}$. The PPM framework makes the assumption that the phonetic event timing distributions are independent of the candidate word duration T, and linearly scales all arrival times in $(t, t+T]$ onto the interval $(0, 1]$ to generate the transformed event set $O'_{t,T}$. The keyword detection function $d_w(t)$, which indicates the presence of the keyword starting at time t with arbitrary duration, is defined as the log-likelihood ratio of phonetic events as described under the keyword and background model. This takes the form

$$\begin{aligned} d_w(t) &= \log\left[\frac{P(O_{t,\infty}|\theta_w)}{P(O_{t,\infty}|\theta_{bg})}\right] \\ &= \log\left[\int_0^\infty \frac{P(O'_{t,T}|T,\theta_w)P(T|\theta_w)}{T^{|O(t)|}P(O_{t,T}|T,\theta_{bg})}dT\right], \end{aligned} \tag{1}$$

where θ_w denotes the set of keyword-specific inhomogeneous Poisson rate parameters, and θ_{bg} denotes background homogeneous rate parameters. Here, the keyword duration T serves as a latent variable with $P(T|\theta_w)$ modeled by a gamma distribution.

3. Extension to Low-Resource Settings

3.1. Deriving Phonetic Events from Low-Resource DNNs

Over the past few years, DNN-HMM hybrid acoustic modeling has become the de facto standard in state-of-the-art speech recognizers. In the context of the Babel program, several groups have attempted to specialize their neural network architectures for limited acoustic training data scenarios [8]. One of our present goals is to evaluate these next-generation acoustic models in the PPM framework based on the assumption that the published word error rate reductions will translate into more accurate phone posterior estimates and, in turn, more accurate phonetic event streams. Now, one of the primary innovations relative to earlier waves of neural networks for ASR is the use of context-dependent HMM state targets. To use these DNNs in the PPM framework, we need to derive monophone posteriorgrams to enable the extraction of the requisite phonetic events. This is easily accomplished by summing together the posterior trajectories of HMM states corresponding to the same context-independent center phone. While we use the DNN trained in the context of an LVCSR system, once we derive monophone posteriorgrams our processing diverges completely from the HMM models and finite state machine based decoders.

Compared with the past neural network phonetic acoustic models [7, 2] evaluated in the PPM framework, our implementation introduces three new components. First, our DNN is trained on top of acoustic features that are speaker adapted with constrained maximum likelihood linear regression (CMLLR), also known as feature-space MLLR (fMLLR) [9]. Note that during training, fMLLR transform estimation is done through computing training alignments using a standard GMM-based, speaker adaptively trained model; in decoding, fMLLR transforms are obtained through first-pass decoding. Thus, for both training alignments and first-pass decoding, the entire knowledge of phonetic context-dependency, pronunciation lexicon and word-level grammar will be integrated, which the single phone recognition system fails to consider.

Second, in addition to basic perceptual linear prediction (PLP) features, we add pitch and probability of voicing (POV) features based on the pitch extraction algorithm described in [10]. Experiments in [10] demonstrate that these pitch and POV features give substantial performance improvements on both tonal and non-tonal languages for LVCSR system, which also contributes to better estimation of phone posteriors. Finally, given the recent success of generalized maxout nonlinear activation functions in DNN modeling, we rely on a DNN acoustic model with p-norm activations [8] of the form $y = \|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{\frac{1}{p}}$, where $\mathbf{x}$ represents a group of neuron inputs. Experiments in [8] demonstrate that DNNs using p-norm units with $p = 2$ perform consistently better than various other nonlinearities evaluated in speech recognition tasks, especially in low-resource conditions.

3.2. Searching for Out-of-Vocabulary Keywords

We consider the KWS task in which keywords are provided in written form in the native orthography and a pronunciation lexicon is given with fixed vocabulary. However, in the low-resource setting a typical condition is that the pronunciation of a given keyword is not covered in the available lexicon. In this case, for the phonetic-based KWS system one standard solution is to predict the pronunciation of OOV keywords by using grapheme-to-phoneme (G2P) conversion [11]. Thus, all OOV keywords become IV and the updated lexicon would contain the phonetic composition of all keywords. However, in the Babel evaluation framework, redecoding the search collection is not allowed after the keywords are known, so other means are required to search using these new predicted pronunciations. Recently, a novel OOV processing technique called proxy keyword search [6] was demonstrated to produce state-of-the-art performance for the task. This method uses the G2P pronunciations of OOV keywords to generate a list of likely-confusable proxy words from the vocabulary. Using a cascade of weighted finite state transducer compositions with the original LVCSR lattice produces putative hits of the OOVs along with lattice posterior confidence scores. Proxy keyword search serves as the baseline OOV method in our experiments.

Using the MAP estimation framework of [4] and given a phonetic pronunciation for an OOV keyword produced by the G2P system, we can construct the dictionary prior PPM. Since we have no examples to estimate the Gaussian parameters within an OOV keyword, we can either assign Gaussian means at equal intervals with fixed variance (based on the simplifying assumption that all phones within the word have equal duration) [4], or estimate the Gaussian parameters for each phone using average phone durations [12]. In this paper, we limit our evaluation to the simple uniform approach, though we would expect the incorporation of average phone duration statistics to provide marginal gains. We further introduce additional Gaussians of likely confused phones that are not in dictionary form using a confusion matrix estimated across entire corpus. More-

over, we apply the Monte Carlo sampling approach explained in [2] to estimate Gamma distribution parameters of each keywords duration model for unseen words. In this way, we can construct a reasonably accurate estimate of PPM rate and word duration parameters without any training exemplars.

3.3. System Combination

We evaluate the combination of the LVCSR and PPM search results by merging the respective putative hit lists. Both system use the identical DNN acoustic model but generate search ranked lists using completely different lexical models and decoding methodologies. The LVCSR system applies HMM lexical models on top of DNN-derived emission likelihoods in a WFST-based decoder that uses a language model. It generates deep word-based lattices that form the search index used for both IV and OOV keywords. The PPM system processes posteriors into an extremely sparse phonetic index and performs a linear-time search. Thus, the system combination evaluation serves to measure the complementarity of these techniques *after* the acoustic processing stages. The resulting putative hit lists from two systems are combined by the following procedure. First, we perform separate score normalization for each using the term-specific threshold technique in [13]. Second, we merge the hits from the two lists that begin and end with less than 0.5 second difference. The combined score for merged hits s_{merge} is computed as

$$s_{\text{merge}} = (w_1 s_1^{1/r} + w_2 s_2^{1/r})^r,$$

where s_1 and s_2 are the individual system scores, w_1 and w_2 are the weights assigned to each system such that $w_1 + w_2 = 1$, and r is a power factor between 1 and 10. The parameters $\{w_i\}$ and r are optimized per language on a development set. Note that given 0-1 normalized input scores, this nonlinear combination rule will produce 0-1 normalized combination scores. Finally, we apply score normalization to the merged hit list.

4. Experiments

4.1. Evaluation Design

We evaluate our PPM KWS performance in the IARPA Babel Program (IARPA-BAA-11-02) framework, which has released conversational telephone speech corpora for several languages. In this study, we measure our system performance on Haitian[1], Lao[2], Assamese[3], Bengali[4] and Zulu[5]. For each language there are two resource conditions: the full language pack (FullLP) contains approximately 80 hours of transcribed speech audio along with a pronunciation dictionary that covers all word types it contains; the limited language pack (LimitedLP) contains a 10 hour subset of FullLP. Language model text and pronunciation dictionary entries for LimitedLP are restricted to those that occur in the given 10 hours. In this paper we only consider LimitedLP, which simulates low-resource conditions for a diverse set of languages. For each language a 10-hour development-testing search collection is also provided to evaluate system performance. Keyword sets are the official development lists generated by Babel participants for use before the evaluation period, which consist of approximately 2000 multi-word queries for each language. We use two KWS scoring metrics. First, Actual Term-Weighted Value (ATWV) [14] is given by

$$ATWV = 1 - \frac{1}{K} \sum_{w=1}^{K} \left(\frac{N_{\text{Miss}}(w)}{N_{\text{True}}(w)} + \beta \frac{N_{\text{FA}}(w)}{T - N_{\text{True}}(w)} \right),$$

where K is the total number of keywords, $N_{\text{Miss}}(w)$ is the number of missed detection of keyword w, $N_{\text{FA}}(w)$ the number of false alarms of w, $N_{\text{True}}(w)$ the number of reference occurrences of w. ATWV requires scores to be both normalized across keyword such that a single global threshold can be set, as well as well calibrated against the true posterior probability of correctness such that the global threshold is 0.5. Second, Oracular Term-Weighted Value (OTWV) is defined assuming the keyword-specific optimal threshold is used instead of 0.5. Since OTWV does not require scores to be normalized across keyword, it is a measure only of ranked list quality. The NIST F4DE scoring tool is used for reference alignment, and YES/NO decisions are made based on posterior scores.

4.2. System Implementation Details

The state-of-the-art DNN infrastructure of the Kaldi toolkit is used as the input phonetic acoustic model. Here, we first train a standard GMM-based, speaker adaptively trained model to obtain HMM-state alignments and fMLLR feature transforms. Next, we train a 5-layer DNN of p-norm units with $p = 2$ [8]. The basic input features are 13-dimensional PLP augmented with 3-dimensional pitch and POV features, and spliced by 3 frames; then the 48-dimensional feature is reduced to 40 dimensions using linear discriminant analysis (LDA). Adaptation with maximum likelihood linear transforms with semi-tied covariance (MLLT/STC) and fMLLR is applied, and 9-frame context windows are stacked to represent the center frame. Thus, the resulting inputs to the DNN are 360 dimensions, and the outputs are posteriors over context dependent HMM-states where the number and identity depend on the language. The current PPM framework operates on monophone posteriorgrams, which are derived by summing posterior dimensions corresponding to the same center phone.

To obtain pronunciations for OOV keywords, we use the Sequitur G2P toolkit [11], a data-driven G2P converter based on joint-sequence models. We use each language's LimitedLP lexicon with pairwise examples of word and pronunciations to train a G2P model, and use the trained model to generate the pronunciation for a given OOV keyword. Each dictionary-based PPM is synthesized according to the prescription given in [4]. For multi-word keywords, we construct the dictionary-based PPM for each unigram in the multi-word keyword, update each unigram PPM if training exemplars for that unigram are available, and then concatenate unigram PPM into a multi-word PPM, as described in [2].

For OTWV calculation, we can use the PPM likelihood ratio detection function directly without tuning any score normalization parameters. However, for the ATWV calculation we must provide confidence scores normalized across keywords. Following [2], we use a simple two-parameter logistic regression (slope and bias) to map PPM detection function scores to posterior probability estimates and apply the term-specific thresholding technique described in [13]. Following [5], we estimate these logistic regression parameters using a 2 hour subset of the 10 hour development set we use for testing. Separately, we performed cross-validation experiments to confirm that this minor train-on-test violation did not unfairly impact our results.

[1]Language collection release `IARPA-babel201b-v0.2b`.
[2]Language collection release `IARPA-babel203b-v3.1a`.
[3]Language collection release `IARPA-babel102b-v0.5a`.
[4]Language collection release `IARPA-babel103b-v0.4b`.
[5]Language collection release `IARPA-babel206b-v0.1e`.

Table 1: *LVCSR, PPM, and combined search performance for the five languages, along with relative gain from combination over the LVCSR baseline alone. Averages are over the corresponding individual language fields.*

Language	System	OTWV (All)	ATWV (All)	ATWV (IV)	ATWV (OOV)
Haitian	LVCSR	0.54	0.44	0.49	0.23
	PPM	0.36	0.21	0.20	0.25
	Comb	0.60	0.48	0.51	0.35
	% Gain	11.1	9.1	4.0	52.2
Lao	LVCSR	0.51	0.41	0.43	0.22
	PPM	0.32	0.16	0.17	0.12
	Comb	0.57	0.44	0.47	0.26
	% Gain	11.8	7.3	9.3	18.2
Zulu	LVCSR	0.28	0.17	0.30	0.09
	PPM	0.27	0.11	0.06	0.14
	Comb	0.41	0.24	0.32	0.19
	% Gain	46.4	41.2	6.7	111.1
Assamese	LVCSR	0.37	0.25	0.31	0.10
	PPM	0.21	0.08	0.08	0.07
	Comb	0.42	0.28	0.34	0.14
	% Gain	13.5	12.0	9.7	40.0
Bengali	LVCSR	0.38	0.27	0.35	0.13
	PPM	0.22	0.10	0.10	0.09
	Comb	0.43	0.30	0.37	0.17
	% Gain	13.2	11.1	5.7	30.8
Averages	LVCSR	0.42	0.31	0.38	0.15
	PPM	0.28	0.13	0.12	0.14
	Comb	0.49	0.35	0.40	0.22
	% Gain	19.2	16.1	7.1	50.5

4.3. Results

Table 1 shows the LimitedLP KWS results on the five languages using the Kaldi LVCSR and PPM systems, as well as the combination of the two. Also listed are the relative fusion gains over the baseline, as well as average performance values over the five languages. Consisten with the results in [2], we find that LVCSR-based search dominates ATWV, with the PPM achieving on average only 42% of the baseline performance. However, we find that PPM search gives much more competitive results on OTWV performance, a metric that evaluates the quality of the ranked list *independent of* the consistency of confidence scores across keywords. This OTWV-ATWV divergence is a consequence of the PPM's suboptimal score normalization, which is performed using a simple logistic regression applied to the likelihood ratio detection score of Eq. 1. Indeed, the LVCSR search system computes true lattice posterior scores, which normalize each lattice arc likelihood by all the other words that might have accounted for the same acoustic observations. This is a much more powerful normalization scheme, but it does come at the larger computational cost of decoding the whole vocabulary at indexing time. For keyword applications that do not require score normalization, the PPM system provides on average 66% of LVCSR baseline OTWV performance with a much smaller index processing time and size (see [2] for details).

If we consider OOV keyword search ATWV in isolation, we can see that the dictionary-based PPM achieves comparable results with the state-of-the-art WFST-based proxy keyword search. The PPM outperforms on Haitian and Zulu, while falling short on Lao, Assamese and Bengali, so it interesting to consider what language-specific properties may be driving this variation. For Zulu, an agglutinative language with a unusually high keyword OOV rate, the PPM system achieves much closer overall KWS performance with LVCSR, indicating PPM's advantage for truly low-resource settings with woefully incomplete pronunciation dictionaries. Note that the PPM usually gives comparable or even higher OOV ATWV results than IV, since we find that PPM search is more sensitive to keyword length and OOV keywords tend to be longer.

Given the distinct lexical modeling strategies employed in the LVCSR baseline and PPM search systems, as well as the substantial relative performance variation across language, some degree of complementarity is to be expected. Even though the PPM overall performance substantially trails the LVCSR baseline on all five languages, we measured a 16% average relative improvement of both ATWV and OTWV in combination. Moreover, the comparable performance of PPMs and proxy keyword search for OOVs combine to produce an average ATWV relative increase of 50% over proxies alone. While in-vocabulary PPM performance lags LVCSR the most, we still post an average relative gain of 7% in fusion.

In terms of runtime comparison between proxy keyword search and PPM OOV search on the 10 hour development set, we compare the average runtime of five languages for the three stages of operation, in terms of CPU time (in seconds). First, for indexing time on the 10 hour search collection, proxy keyword search takes 5,736 seconds to make an inverted index from decoding lattices, while the PPM system takes 256 seconds to extract phonetic events from monophone posteriorgrams. Second, for model construction, it takes 2.4 seconds to generate word proxies for each keyword, while it takes 0.01 seconds to construct one dictionary prior PPM. Finally, for searching the index, proxy search takes 0.55 seconds for each keyword, while the PPM search takes 0.08 seconds (computed using the benchmark information provided in [2]). In all three categories, we find that OOV search with PPMs is significantly more efficient in time than proxy keyword search. It does require an additional phone event index, but as demonstrated in [2], the index construction time and size are negligible.

5. Conclusions

We have demonstrated that the point process model framework provides a viable keyword search platform for low-resource settings. It is highly complementary with state-of-the-art LVCSR techniques, posting substantial fusion gains for every language evaluated. On its own, it provides state-of-the-art handling of OOV keywords, but also produces dramatics gains when combined with proxy keyword search outputs. However, as evidenced by comparatively large gaps between ATWVs and OTWVs, the substandard score normalization achievable with PPMs remains a major challenge. Thus, the incorporation of competing hypotheses and contextual constraints into the PPM search is the main avenue for future progress.

6. Acknowledgments

The authors were supported in part by IARPA Babel contract No. W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. 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 DARPA, IARPA, DoD/ARL or the U.S. Government.

7. References

[1] Aren Jansen and Partha Niyogi, "Point process models for spotting keywords in continuous speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 8, pp. 1457–1470, 2009.

[2] Keith Kintzley, Aren Jansen, and Hynek Hermansky, "Featherweight phonetic keyword search for conversational speech," in *ICASSP*, 2014.

[3] Aren Jansen and Partha Niyogi, "Detection-based speech recognition with sparse point process models," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4362–4365.

[4] Keith Kintzley, Aren Jansen, and Hynek Hermansky, "MAP estimation of whole-word acoustic models with dictionary priors," in *INTERSPEECH*, 2012.

[5] Guoguo Chen, Sanjeev Khudanpur, Daniel Povey, Jan Trmal, David Yarowsky, and Oguz Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in lowresource languages," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8560–8564.

[6] Guoguo Chen, Oguz Yilmaz, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 416–421.

[7] Keith Kintzley, Aren Jansen, and Hynek Hermansky, "Event selection from phone posteriorgrams using matched filters.," in *INTERSPEECH*, 2011, pp. 1905–1908.

[8] Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*, 2014.

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

[10] Pegah Ghahremani, Bagher BabaAli, Daniel Povey, Korbinian Riedhammer, Jan Trmal, and Sanjeev Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *ICASSP*, 2014.

[11] Maximilian Bisani and Hermann Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[12] Keith Kintzley, Aren Jansen, and Hynek Hermansky, "Text-to-speech inspired duration modeling for improved whole-word acoustic models," in *INTERSPEECH*, 2013.

[13] David RH Miller, Michael Kleber, Chia-Lin Kao, Owen Kimball, Thomas Colthurst, Stephen A Lowe, Richard M Schwartz, and Herbert Gish, "Rapid and accurate spoken term detection.," in *INTERSPEECH*, 2007, pp. 314–317.

[14] NIST, "The Spoken Term Detection (STD) 2006 Evaluation Plan," `http://www.nist.gov/speech/tests/std/`, 2006.