A KEYWORD-AWARE GRAMMAR FRAMEWORK FOR LVCSR-BASED SPOKEN KEYWORD SEARCH

I-Fan Chen[1], Chongjia Ni[2], Boon Pang Lim[2], Nancy F. Chen[2], and Chin-Hui Lee[1]

[1]School of Electrical and Computer Engineering, Georgia Institute of Technology
[2]Institute for Infocomm Research, Singapore

ABSTRACT

In this paper, we proposed a method to realize the recently developed keyword-aware grammar for LVCSR-based keyword search using weight finite-state automata (WFSA). The approach creates a compact and deterministic grammar WFSA by inserting keyword paths to an existing *n*-gram WFSA. Tested on the evalpart1 data of the IARPA Babel OpenKWS13 Vietnamese and OpenKWS14 Tamil limited-language pack tasks, the experimental results indicate the proposed keyword-aware framework achieves significant improvement, with about 50% relative actual term weighted value (ATWV) enhancement for both languages. Comparisons between the keyword-aware grammar and our previously proposed *n*-gram LM based approximation approach for the grammar also show that the KWS performances of these two realizations are complementary.

Index Terms— keyword search, spoken term detection, grammar network, weighted finite-state automaton

1. INTRODUCTION

Spoken keyword search (KWS) [1, 2] is a task of detecting a set of preselected keywords in continuous speech. The technology has been used in various applications, such as spoken term detection [3-8], spoken document indexing and retrieval [9], speech surveillance [10], spoken message understanding [11, 12], etc. In general, KWS systems can be categorized into two groups depending on grammars[1] used by the systems: (i) classic keyword-filler based KWS [1, 2], and (ii) large vocabulary continuous speech recognition (LVCSR) based KWS [3-7].

In the classic keyword-filler based KWS, a spoken utterance is represented as a sequence of keywords and non-keywords (often referred to as fillers [1]), and the decoding grammar is a simple keyword-filler loop network (Fig. 1. (a)) [1, 2]. Because of the simplicity, a keyword-filler based system often achieves a high detection rate using only a small amount of data for acoustic model training. However the systems are restricted to the set of predefined keywords and often produce a great amount of false alarms which requires follow-up utterance verification [13-16] to decide if the detected keyword segments are true hits or false alarms.

On the other hand, LVCSR-based KWS first converts speech utterances into word or sub-word level text documents using speech-to-text (STT) techniques [17-19] with *n*-gram language model (LM) [20] based grammars (Fig. 1. (b)) and then perform keyword search on the STT-transcribed documents [3-7]. This allows the KWS systems to search any keyword without reprocessing the speech signal. In general, LVCSR-based KWS gives better detection results and much fewer false alarms than keyword-filler based KWS [21] because more linguistic information is utilized in the framework. However, since a high-performance *n*-gram language model typically requires a significant amount of text training data [22, 23], it often becomes a major performance bottleneck with more detection misses for LVCSR-based KWS in applications where only limited linguistic resources are available. This is indeed a major issue when language models can only been trained from a limited set of transcribed audio data.

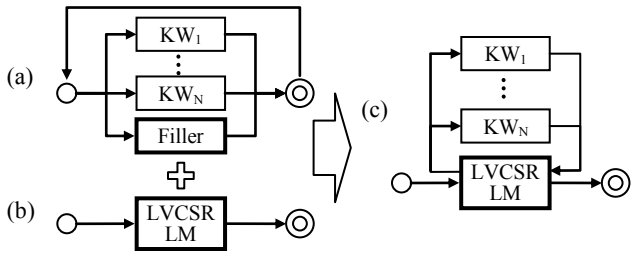

Fig. 1. Illustration of (a) keyword-filler loop grammar, (b) LVCSR LM grammar, and (c) the proposed keyword-aware grammar for KWS

Recently, we found that by adopting a keyword-aware (KW-aware) grammar framework, which integrates the keyword-filler loop grammar into the *n*-gram LM grammar used by LVCSR-based KWS (as illustrated in Fig. 1 (c)), we could achieve significant performance improvement for both poorly-trained and well-trained *n*-gram LMs [24]. Preliminary results using context-simulated keyword language model (CS-KWLM) interpolated LMs [24], which approximate the effect of the new grammar, reveal that the proposed framework not only preserves the characteristics

[1] In this paper, a grammar is defined as a search graph or network whose paths from initial to final nodes represent valid word sequences in the system with corresponding scores; such a grammar can be easily realized by a weighted finite-state automaton (WFSA) [15].

of high accuracy, low false alarms, and keyword flexibility of LVCSR-based KWS, but also inherits the high detection rate from keyword-filler based KWS under resource-limited conditions. In this paper, we show how to implement the KW-aware grammar with weighted finite-state automata (WFSA) without any approximation. We also compare the WFSA-realized KW-aware grammar with the previously proposed CS-KWLM interpolated LM approximation to show the similarities and differences between the two realizations.

2. KEYWORD-AWARE GRAMMAR

When *n*-gram LMs are poorly trained with limited or topic-mismatched data, LVCSR-based KWS usually suffers from high detection misses due to underestimated keyword prior probabilities. To alleviate the problem, in the KW-aware grammar framework, probabilities of keywords are boosted by inserting additional standalone keyword [2] paths with appropriate scores to the *n*-gram LM grammars. Thus even when the *n*-gram LMs are poorly trained the system can still have reasonable prior probability estimation for the target keywords. The conditional probability of a keyword k (k=$w_1 \ldots w_L$) with context history h in the KW-aware grammar is therefore

$$P_{KW-aware}(k \mid h) = \max\left\{ P_{n-gram}(k \mid h) \ , \ \kappa \right\}, \quad (1)$$

where $P_{n-gram}(k \mid h) = \prod_{i=1}^{L} P(w_i \mid h_i)$ is the probability estimated by regular *n*-gram LMs, and κ is a prior constant as the weight of the additional path for the query k to control the minimal value of $P_{KW\text{-}aware}(k|h)$. In real task, we may simply use a global prior constant, κ, for all the system keywords, or we can categorize the keywords into a set of classes with their own prior constants. The κ(s) is(are) the parameter(s) to be tuned in the grammar.

Note that by default in the KW-aware grammar a keyword can be represented by either the *n*-gram LM or a keyword path, which makes the grammar nondeterministic and ineligible for offline optimization. In the next section, we will show how to use disambiguation symbols to solve the problem.

3. REALIZATION OF THE GRAMMAR

In this section, we first briefly introduce the WFSA representation of *n*-gram LMs [25, 26]. Then we show how a deterministic KW-aware grammar WFSA can be realized by modifying the *n*-gram LM WFSA. An approximation approach we proposed previously is also introduced. For WFSA formulations, annotations in [26] are adopted in this paper.

3.1. Preliminary

Definition 1. *A system* $(\mathbb{K}, \oplus, \otimes, \bar{0}, \bar{1})$ *is a* semiring [27] *if*: $(\mathbb{K}, \oplus, \bar{0})$ *is a commutative monoid with identity element* $\bar{0}$; $(\mathbb{K}, \otimes, \bar{1})$ *is a monoid with identity element* $\bar{1}$; $\otimes$ *distributes over* $\oplus$; *and* $\bar{0}$ *is an annihilator for* $\otimes$: *for all* $a \in \mathbb{K}$, $a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$.

In this paper, the log semiring $\mathcal{L}=(\mathbb{R} \cup \{\infty\}), \oplus_{\log}, +, \infty, 0)$ is used [25]. Note the log semiring is an isomorphism of the probability semiring $(\mathbb{R}_+, +, \times, 0, 1)$ via a log morphism with, for all $a, b \in \mathbb{R} \cup \{\infty\}$:

$$a \oplus_{\log} b = -\log(\exp(-a) + \exp(-b))$$

and we follow the convention that $\exp(-\infty)=0$ and $-\log(0)=\infty$.

Definition 2 *A weighted finite-state automaton A over a semiring* $\mathbb{K}$ *is an 7-tuple* $A=(\Sigma, Q, I, F, E, \lambda, \rho)$ *where*: Σ *is the finite alphabet of the automaton*; Q *is a finite set of states*; $I \subseteq Q$ *the set of initial states*; $F \subseteq Q$ *the set of final state*; $E \subseteq Q \times (\Sigma \cup \{\infty\}) \times \mathbb{K} \times Q$ *a finite set of transitions*; λ: $I \rightarrow \mathbb{K}$ *the initial weight function*; *and* ρ: $F \rightarrow \mathbb{K}$ *the final weight function mapping F to* $\mathbb{K}$.

Given a transition $e \in E$, we denote its label $l[e]$, its origin or previous state $p[e]$ and its destination state or next state $n[e]$, its weight $w[e]$, namely $e=(p[e], l[e], w[e], n[e])$. Given a state $q \in Q$, we denote by $E[q]$ the set of transitions leaving q.

A path $\pi=e_1 \ldots e_L$ is an element of E^* with consecutive transitions: $n[e_{i-1}] = p[e_i]$, $i=2, \ldots, L$. We extend n and p to paths by setting $n[\pi] = n[e_L]$ and $p[\pi] = p[e_1]$. The labeling function l and the weight function w can also be extended to paths by defining the label of a path as a concatenation of the labels of its constituent transitions, and the weight of the path as the $\otimes$-product of the weights of its constituent transitions: $l[\pi]=l[e_1] \ldots l[e_L]$, $w[\pi]=w[e_1] \otimes \ldots \otimes w[e_L]$. The path π can therefore be represented by $(p[\pi], l[\pi], w[\pi], n[\pi])$. We also define *states*$[\pi]$ and *transitions*$[\pi]$ being the set of states and transitions on the path π.

3.2. Representation of *n*-gram LMs with WFSAs

In a WFSA representation of an *n*-gram LM over the log semiring, each state in the WFSA represents an *n*-gram conditioning history h_i, e.g. $w_{i-2}w_{i-1}$. Each transitions leaving the state represent a word w_i with a weight $-\log(P(w_i|h_i))$ or a backoff transition to a lower-order conditioning history state [26]. A string accepted by the WFSA has a single path through the automaton, and the weight of the string is the sum of the transition weights in that path in a form of negative log probability.

Given a finite set of state, Q, in an *n*-gram WFSA and a string $k=w_1 \ldots w_L$, we denote *hist of*$[k, Q]$ as the state in Q encoding the conditioning history that matches the end of the string k with the highest order.

3.3. Realization of KW-aware grammar with WFSAs

Suppose the set of system keywords can be categorized into c classes, and K_i (i=1~c) is a list of keywords in class i with the list size $|K_i|$ and the constant prior κ_i for the class, given

[2] Note keywords in this paper refer to single- or multi-word queries.

an n-gram LM WFSA, $A=(\Sigma, Q, I, F, E, \lambda, \rho)$, a KW-aware grammar WFSA, A', can be realized by the pseudo code presented in Fig. 2. The algorithm consists of four steps: (i) add disambiguation symbols to the alphabet of WFSA, (ii) add keyword initial states, (iii) add keyword paths, and (iv) normalization to make the final KW-aware WFSA stochastic. Note that in the KW-aware grammar WFSA we utilize disambiguation symbols ($\#\boldsymbol{k_1}$, …, $\#\boldsymbol{k_c}$) on any transition from states in the n-gram WFSA to the keyword initial states (line 7). The resulting WFSA is therefore deterministic and can be optimized offline. In this paper, all keywords are assumed in the same class, and a single keyword initial state and κ are used.

Create KW-aware grammar WFSA (A, $K_{1\sim c}$, $\kappa_{1\sim c}$)
1 $A' \leftarrow A$
2 $\Sigma' \leftarrow \Sigma \cup \{\#\boldsymbol{k_1}, \ldots, \#\boldsymbol{k_c}\}$ // 1. Add disambiguation symbols
3 **for** i **in** 1 **to** c **do**:
4 $q_{ki} \leftarrow (K\_Init_i)$
5 $Q' \leftarrow Q' \cup \{q_{ki}\}$ // 2. Add keyword initial states
6 **for** $q \in Q$ **do** :
7 $E' \leftarrow E' \cup \{(q, \#\boldsymbol{k_i}, -\log(|K_i|\cdot\kappa_i), q_{ki})\}$
8 **for** $k \in K_i$ **do** : // 3. Add keyword paths
9 $\pi \leftarrow (q_{ki}, k, \log(|K_i|), hist\_of[k, Q])$
10 $Q' \leftarrow Q' \cup states(\pi)$
11 $E' \leftarrow E' \cup transitions(\pi)$
12 **for** $q' \in Q'$ **do**: // 4. Normalization
13 $norm \leftarrow \oplus_{e' \in E[q']} w[e']$
14 **for** $e' \in E[q']$ **do**:
15 $e' \leftarrow (q', l[e'], w[e'] - norm, n[e'])$
16 **return** A'

Fig. 2. Pseudo code for the KW-aware grammar WFSA realization.

3.4. Approximation of the KW-Aware grammar

In [24], the boosting effect of Eq. (1) is approximated by interpolating the original n-gram LM with a keyword LM. The training text of the keyword LM consists of the system keywords prefixed and suffixed by common context terms derived from the original training text. This context-simulated keyword LM (CS-KWLM) has been shown to provide significant performance enhancement for KWS systems [24]. Eq. (2) is the interpolation formula for the two LMs, and α is an interpolation weight needed to be tuned.

$$P_{INT\_LM}(w|h) = \alpha \cdot P_{CS-KWLM}(w|h) + (1-\alpha) P_{LM}(w|h). \quad (2)$$

4. EXPERIMENTAL SETUP

Experiments were conducted on the IARPA Babel OpenKWS13 (Vietnamese) [28] and OpenKWS14 (Tamil) [29] limited language pack (LLP) tasks[3]. In both tasks only 10-hour transcribed audio were used for system training. The data are conversational speech between two parties over a telephone channel, which can be landline, cellphone, or phones embedded in vehicles, with the sampling rate set at 8000 Hz. For system tuning, we used the 10-hour IARPA development sets (denoted as dev10h) for each language.

For both OpenKWS13 and OpenKWS14 systems, the 15-hour evaluation part 1 data (released as evalpart1 by NIST) were used for testing. The evaluation keyword lists contain 4065 and 5576 phrases with out-of-vocabulary words not appearing in the training set for the two tasks respectively. The performance of keyword search was measured by the number of missing keywords and the actual term weighted value (ATWV) [30], which is a metric that takes both detection miss and false alarm errors into account. A system with perfect detection performance would have ATWV of 1. Note that the IARPA Babel program set ATWV=0.3 as the benchmark for KWS system performance.

All keyword search systems were LVCSR-based with hybrid DNN-HMM acoustic models built with the Kaldi toolkit [31]. Readers can easily reproduce all baseline results presented in this paper by running the Babel recipe provided in the Kaldi toolkit. The DNNs were trained with sMBR sequential training [32]. The acoustic features were bottleneck features appended with fMLLR features, while the bottleneck features were built on top of a concatenation of PLP, fundamental frequency (F0) features, and for the Vietnamese systems fundamental frequency variation (FFV) features were used in the concatenation as well. We used a grapheme-to-phoneme (G2P) approach [33] to estimate the pronunciation for OOV words appearing in the evaluation keywords. The estimated pronunciations were then merged into the original lexicon provided by IARPA to form the system lexicon.

Three KWS systems were compared. While all the systems shared the same acoustic model and lexicon, they are different in the decoding grammars. The first system (denoted as "n-gram baseline") is the baseline which used the original trigram LM. The second system is the proposed keyword-aware grammar based system (denoted as "KW-aware grammar") with a global prior constant κ. And the third used the approximate CS-KWLM Interpolated LM as system grammar (denoted as "CS-KWLM Int"). All the system parameters α and κ are tuned with the dev10h data for each task.

5. EXPERIMENTAL RESULTS AND DISCUSSION

5.1. Comparison of grammar WFSAs

Table I compares the Vietnamese grammar WFSAs used in the three systems. Carrying additional keyword information, both KW-aware grammar and CS-KWLM Int systems have larger grammar WFSAs than the baseline. However, as the size of the CS-KWLM based grammar WFSA being 10 times larger than the baseline due to the great amount of additional keyword n-gram states, the size of the KW-aware grammar remained in a similar scale of the original n-gram WFSA. It is clear that the exact realization provides more compact grammar WFSA than the approximate approach.

[3] This study uses the IARPA Babel Program Vietnamese and Tamil language collection releases babel107b-v0.7 and IARPA-babel204b-v1.1b with the LimitedLP training sets.

Table I. Vietnamese grammar WFSAs used in the three systems

Vietnamese grammars	# arcs	# states	File size
n-gram baseline	38,713	17,616	812 Kb
KW-aware grammar (global κ=0.00005)	66,913	24,215	1.3 Mb
CS-KWLM Int (α=0.6)	381,461	165,063	7.8 Mb

5.2. System performance

Table II compares performance of the three systems on the Vietnamese evalpart1 data. The baseline system had 2,562 missing keywords and was with ATWV of 0.2098. With the keyword-aware framework, both CS-KWLM and KW-aware grammar systems significantly reduced the number of missed keyword by roughly 40%. The significant reduction of misses also reflected on the improvement of ATWV. The ATWV for KW-aware grammar achieved 0.3224, which is about a 53.7% relative improvement over the baseline. Note that from the averaged ATWV over the all keywords, the CS-KWLM system seemed to perform slightly better than the KW-aware grammar system. However, in the next section we will show that the two realizations are good at detecting different types of keywords.

Table II. Performance on the Vietnamese LLP evalpart1 data.

Vietnamese [evalpart1]		# Miss	ATWV
n-gram baseline		2562	0.2098
KW-aware framework	KW-aware grammar (global κ=0.00005)	1589	0.3224
	CS-KWLM Int (α=0.6)	1651	0.3287

Similar trend were found in the Tamil LLP task. In Table III, the baseline system had 3,663 missed keywords and ATWV of 0.2128. Again, the KW-aware framework reduced about one third of the miss in the baseline. And a relative 46% ATWV improvement was also observed on the KW-aware grammar and CS-KWLM systems.

Table III. Performance on the Tamil LLP evalpart1 data.

Tamil [evalpart1]		# Miss	ATWV
n-gram baseline		3663	0.2128
KW-aware framework	KW-aware grammar (global κ=0.0000347)	2830	0.3102
	CS-KWLM Int (α=0.3)	2689	0.3160

5.3. ATWV analysis for keywords of different lengths

To further study the characteristics of each system, we compared performances of the three systems on keywords of different lengths. Fig. 3 displays the ATWV curves for the *n*-gram baseline, CS-KWLM and the KW-aware grammar systems in the Tamil LLP task. In general, a KWS system has better detection performance for longer keywords because more acoustic context information is available for the system to make correct decisions. However, because of the misses caused by the underestimated keyword priors, the ATWVs of the *n*-gram baseline system in Fig. 3 only increased slowly with the increase of keyword lengths and dropped rapidly when keyword length $L > 3$. By alleviating the underestimation problem, both CS-KWLM and KW-aware grammar systems significantly outperformed the baseline system, especially for long keywords.

If we further compare the CS-KWLM and KW-aware grammar systems, it is clear that the two systems were different in their performance with the keyword lengths. For long keyword (L>2), the KW-aware grammar significantly outperformed the CS-KWLM system. This is because in the CS-KWLM approach we not only boost probabilities of keywords but also other word sequences with keyword *n*-grams presented. The boosting effect may therefore being reduced relatively especially for multi-word keywords because more *n*-grams are presented in the queries. The standalone keyword paths in KW-aware grammar to some extent alleviate this problem caused by *n*-gram sharing in the *n*-gram LM, thus it has better performance for long keywords. On the other hand, the CS-KWLM system has slightly better performance than KW-aware grammar for $L \leq 2$ because keyword priors in the KW-aware grammar system were restricted to a global κ while for the CS-KWLM system such restriction did not exist which allowed the prior estimation for each keyword being closer to its ground truth. Since keywords with $L \leq 2$ are the majority in the evaluation list, the overall ATWV of CS-KWLM system in Table III is slightly better than KW-aware grammar system. However, the result suggests that the two realizations are complementary and should be considered in different cases.

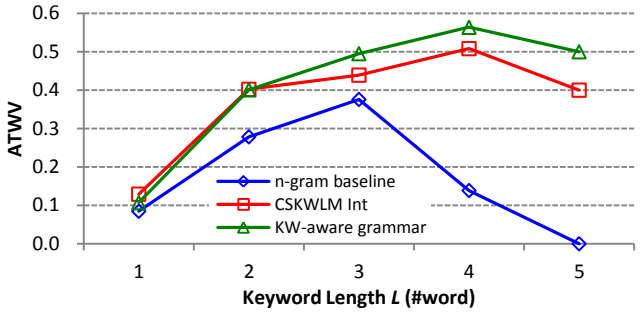

Fig. 3. ATWV of keywords with different lengths for the three systems on evalpart1 data in the Tamil LLP task. The ATWV drops at L=5 in all the systems are due to miss errors caused by underestimated keyword priors.

6. CONCLUSION

In this paper, we proposed an exact realization, which was a missing block in the current KW-aware framework, of the KW-aware grammar. Experimental results on Babel Vietnamese and Tamil LLP tasks show the exact realization was very compact and outperformed our previous approximation method for long keywords; while for short keywords, the performances of the two systems were similar. We also showed that the significant performance improvement of the proposed KW-aware framework over the *n*-gram baseline is consistent across languages. The complementary performances of the two realizations for keywords with different lengths also suggest us that a combination of the two realization methods might bring further improvement to the LVCSR-based KWS systems.

7. REFERENCES

[1] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 38, pp. 1870-1878, 1990.

[2] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," in *Proc. ICASSP-90*, 1990, pp. 129-132.

[3] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark*, et al.*, "The SRI/OGI 2006 spoken term detection system," in *Proc. Interspeech*, 2007, pp. 2393-2396.

[4] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proceedings of the 30 th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 615-622.

[5] D. R. Miller, M. Kleber, C.-l. Kao, O. Kimball, T. Colthurst, S. A. Lowe*, et al.*, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007.

[6] R. Wallace, R. Vogt, and S. Sridharan, "A Phonetic Search Approach to the 2006 NIST Spoken Term Detection Evaluation," in *Proc. Interspeech*, 2007.

[7] N. F. Chen, S. Sivadas, B. P. Lim, H. G. Ngo, H. Xu, V. T. Pham*, et al.*, "STRATEGIES FOR VIETNAMESE KEYWORD SEARCH," in *Proc. ICASSP*, 2014.

[8] N. F. Chen, C. Ni, I.-F. Chen, S. Sivadas, V. T. Pham, H. Xu*, et al.*, "LOW-RESOURCE KEYWORD SEARCH STRATEGIES FOR TAMIL," in *Proc. ICASSP*, 2015.

[9] J. Makhoul, F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz*, et al.*, "Speech and Langauge Technologies for Audio Indexing and Retrieval," *Proc. IEEE,* vol. 88, pp. 1338-1353, 2000.

[10] R. L. Warren., "BROADCAST SPEECH RECOGNITION SYSTEM FOR KEYWORD MONITORING," U.S. Patent 6332120 B1, 2001.

[11] T. Kawahara, C.-H. Lee, and B.-H. Juang, "Key-Phrase Detection and Verification for Flexible Speech Understanding," *IEEE Trans. on Speech and Audio Proc.,* vol. 6, pp. 558-568, Nov. 1998.

[12] B.-H. Juang and S. Furui, "Automatic Recognition and Understanding of Spoken Language – A First Step Toward Natural Human-Machine Communication," *Proc. IEEE,* vol. 88, pp. 1142-1165, Aug. 2000.

[13] R. A. Sukkar and C.-H. Lee, "Vocabulary Independent Discriminative Utterance Verification for Non-Keyword Rejection in Subword Based Speech Recognition," *IEEE Trans. on Speech and Audio Proc.,* vol. 4, pp. 420-429, Nov. 1996.

[14] M. Rahim, C.-H. Lee, and B.-H. Juang, "Discriminative Utterance Verification for Connected Digit Recognition," *IEEE Trans. on Speech and Audio Proc.,* vol. 5, pp. 266-277, May 1997.

[15] M. G. Rahim, L. Chin-Hui, and J. Biing-Hwang, "Discriminative utterance verification for connected digits recognition," *Speech and Audio Processing, IEEE Transactions on,* vol. 5, pp. 266-277, 1997.

[16] M.-W. Koo, C.-H. Lee, and B.-H. Juang, "Speech Recognition and Utterance Verification Based on a Generalized Confidence Score," *IEEE Trans. on Speech and Audio Proc.,* vol. 9, pp. 821-832, Nov. 2001.

[17] C.-H. Lee, F. K. Soong, and K. K. Paliwal, Eds., *Automatic Speech and Speaker Recognition: Advanced Topics*. Kluwer Academic Publishers, 1996.

[18] F. Jelinek, *Statistical Method for Speech Recognition*: MIT Press, 1997.

[19] J.-L. Gauvain and L. Lamel, "Large Vocabulary Continuous Speech Recognition: Advances and Applications," *Proc. IEEE,* vol. 88, pp. 1181-1200, August 2000.

[20] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language,* vol. 13, pp. 359-393, October 1999 1999.

[21] I. Szoke, P. Schwarz, P. Matejka, L. Burget, M. Karafiat, M. Fapso*, et al.*, "Comparison of Keyword Spotting Approaches for Informal Continuous Speech," in *EuroSpeech*, 2005.

[22] R. Rosenfeld, "Two Decades of Statistical Language Modeling: Where Do We Go from Here?," *Proc. IEEE,* vol. 88, pp. 1270-1278, 2000.

[23] P. Jeanrenaud, E. Eide, U. Chaudhari, J. McDonough, K. Ng, M. Siu*, et al.*, "Reducing word error rate on conversational speech from the Switchboard corpus," in *Proc. ICASSP-95*, 1995, pp. 53-56 vol.1.

[24] I.-F. Chen, C. Ni, B. P. Lim, N. F. Chen, and C.-H. Lee, "A Novel Keyword+LVCSR-FIller Based Grammar Network Representation for Spoken Keyword Search," in *Proc. ISCSLP*, Singapore, 2014.

[25] M. Mohri, F. Pereira, and M. Riley, "Speech Recognition with Weighted Finite-State Transducers," in *Springer Handbook of Speech Processing*, ed: Springer Berlin Heidelberg, 2008, pp. 559-584.

[26] C. Allauzen, M. Mohri, and B. Roark, "Generalized algorithms for constructing language models," in *the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 40-47.

[27] W. Kuch and A. Salomaa, Eds., *Semirings, automata, languages*. London, UK: Springer-Verlag, 1986.

[28] *NIST Open Keyword Search 2013 Evaluation (OpenKWS13)*. Available: http://www.nist.gov/itl/iad/mig/openkws13.cfm

[29] *NIST Open Keyword Search 2014 Evaluation (OpenKWS14)*. Available: http://www.nist.gov/itl/iad/mig/openkws14.cfm

[30] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddintion, "Results of the 2006 Spoken Term Detection Evaluation," in *Proc. SIGIR*, 2007.

[31] D. Povey, A. Ghoshal, G. Boulianne, L. s. B. ˇ, O. r. Glembek, N. Goel*, et al.*, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, ed Hilton Waikoloa Village, Big Island, Hawaii, US: IEEE Signal Processing Society, 2011.

[32] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative traning of deep neural networks," in *Proc. Interspeech*, Lyon, France, 2013.

[33] J. R. Novak, N. Minematsu, and K. Hirose, "WFST-based Grapheme-to-Phoneme Conversion: Open Source Tools for Alignment, Model-Building and Decoding," in *Proc. International Workshop on Finite State Methods and Natural Language Processing*, Donostia-San Sebastian, 2012, pp. 45-49.