

Building A Vocabulary Self-Learning Speech Recognition System

Long Qin¹, Alexander Rudnicky²

¹M*Modal, 1710 Murray Ave, Pittsburgh, PA, USA

²Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, USA

long.qin@mmodal.com, alex.rudnicky@cs.cmu.edu

Abstract

This paper presents initial studies on building a vocabulary self-learning speech recognition system that can automatically learn unknown words and expand its recognition vocabulary. Our recognizer can detect and recover out-of-vocabulary (OOV) words in speech, then incorporate OOV words into its lexicon and language model (LM). As a result, these unknown words can be correctly recognized when encountered by the recognizer in future. Specifically, we apply the word-fragment hybrid system framework to detect the presence of OOV words. We propose a better phoneme-to-grapheme (P2G) model so as to correctly recover the written form for more OOV words. Furthermore, we estimate LM scores for OOV words using their syntactic and semantic properties. The experimental results show that more than 40% OOV words are successfully learned from the development data, and about 60% learned OOV words are recognized in the testing data.

Index Terms: Vocabulary learning, OOV word detection and recovery, lexicon expansion

1. Introduction

Most speech recognition systems are closed-vocabulary and do not accommodate out-of-vocabulary (OOV) words. But in many applications, e.g., *voice search* or *spoken dialog systems*, OOV words are usually content words such as names and locations which contain information crucial to the success of these tasks. Speech recognition systems in which OOV words can be detected and recovered are therefore of great interest.

Several approaches are recently proposed for OOV word detection [1-17], where the word-fragment hybrid system receives the most attention. Hybrid speech recognition systems apply a hybrid lexicon and hybrid language model (LM) during decoding to explicitly represent OOV words with smaller sub-lexical units [9-17]. In our previous work, we built hybrid systems for OOV word detection and recovery using mixed types of sub-lexical units [18-20]. We also studied how to identify recurrent OOV words and how to use their multiple occurrences to estimate lexical properties for OOV words [21, 22].

In this paper, we extend our previous work to design a speech recognition system that can automatically learn new words during its routine operation. To achieve this goal, we need to convert OOV words into IV words. In particular, we need to estimate the pronunciation and written form for OOV words, so that we can integrate OOV words into the recognizer's lexicon. We also need to estimate LM scores for OOV words to add them into the LM. First, we obtain the pronunciation of OOV words from the hybrid decoding result. Then we estimate the written form of OOV words through the phoneme-to-grapheme (P2G) conversion, where we train a better P2G model by learning from recognition errors on training speech. Finally,

we estimate LM scores for OOV words using their syntactic and semantic properties. Particularly, we estimate LM scores of an OOV word by utilizing its part-of-speech (POS) label. We also predict new contexts an OOV word may appear in future based on its semantic relatedness with IV words. The proposed work is tested on tasks with different speaking styles and recording conditions including the Wall Street Journal (WSJ), Broadcast News (BN), and Switchboard (SWB) datasets. Our experimental results show that we can learn more than 40% OOV words from the development data and recognize about 60% learned OOV words in the testing data.

The remainder of this paper is organized as follows. Section 2 describes major components of our recognition system, including the word-fragment hybrid system framework, the lexicon expansion module, as well as the LM score estimator. Section 3 and 4 discuss experiments and results. Concluding remarks are provided in Section 5.

2. Method

Building a vocabulary self-learning speech recognition system involves several steps. We first detect the presence of OOV words in speech using the word-fragment hybrid system. From the hybrid decoding result, we collect the pronunciation for each OOV word and then perform P2G conversion to infer its written form. At this point, we can integrate recovered OOV words into the recognition lexicon. To add these words into the LM, we estimate LM scores for seen OOV n -grams using the inferred POS label of an OOV word. We also predict unseen contexts where OOV words may appear in future based on their semantic relatedness with in-vocabulary (IV) words.

2.1. OOV word detection using the hybrid system

We use a hybrid lexicon and hybrid LM during decoding to detect the presence of OOV words. The hybrid lexicon is obtained by incorporating sub-lexical units and their pronunciations into the word lexicon. The hybrid LM is trained in a flat manner. First, the pronunciation of OOV words is estimated through the grapheme-to-phoneme (G2P) conversion [23], and then used to train the sub-lexical units. After that, OOV words in the training text are replaced by corresponding sub-lexical units to get a new hybrid text corpus. Finally, a hybrid LM is trained from this hybrid text data. When training the hybrid LM, sometimes two or more OOV words may appear consecutively in the training data. After representing OOV words using sub-lexical units, we lose the word boundary between two OOV words. To solve this problem, we add two more symbols into the sub-lexical sequence of each OOV word, which are the word start “^” and word end “\$”. In this paper, we use a word-syllable hybrid system which allows us to achieve the best OOV word recovery

performance. More details of our hybrid system can be found in [18, 20].

In the hybrid decoding result, we consider recognized syllable unit sequences as detected OOV words, where word boundary symbols are used to segment a sequence of syllable units into multiple OOV words. We then extract the pronunciation for OOV words by concatenating their corresponding syllable units. Since recognition errors may be embedded in the decoded syllable units, the estimated pronunciation of an OOV word may contain errors. For example, as given in Table 1, the correct pronunciation of OOV word w_3 should be “K AE D R IY”. The quality of estimated OOV word pronunciations will depend on the hybrid decoding accuracy.

Table 1: Examples of the estimated pronunciation of detected OOV words.

Detected OOV	Estimated Pronunciation
w_1	N OW L AH N
w_2	HH AO T EH N S
w_3	K AE D IY

2.2. Incorporating OOV words into lexicon

After extracting the pronunciation for detected OOV words from the hybrid decoding result, we perform the P2G conversion to produce a written form. To achieve better P2G conversion performance, we train a 6-gram joint sequence model with short grapheme units as suggested in [24]. With both the pronunciation and written form of an OOV word, we can then integrate it into the lexicon.

In our experiment, we found that the estimated pronunciation for many OOV words is incorrect. But the P2G model, which is trained using only IV words whose pronunciation is always correct, cannot handle phone sequences embedded with recognition errors. As a result, we are unable to estimate the correct written form for many detected OOV words. Figure 1 shows the P2G conversion on OOV word w_3 . It can be seen that the estimated spelling “CADY” is wrong, because w_3 is incorrectly recognized as “K AE D IY”.

	Pronunciation		Spelling
HYP	K AE D IY		CADY
REF	K AE D R IY		CADRE

Figure 1: An example of the P2G conversion on an OOV word with incorrect pronunciation.

We therefore investigated training a better P2G model using both positive and negative examples, so that we can correctly estimate the written form of an OOV word even if its pronunciation includes recognition errors. Positive examples are alignments between the correct spelling and correct pronunciation collected from IV words, while negative examples are alignments between the correct spelling and incorrect pronunciation collected from the hybrid decoding results of the training speech. We assume that the recognizer will make similar errors on the training speech as on the testing speech, therefore our P2G model should learn rules to match incorrect pronunciations to correct spellings. The P2G conversion should more

accurately recover OOV orthographies from the noisy hybrid decoding output. For instance, we can now estimate the correct written form “CADRE” for OOV word w_3 from its incorrect pronunciation “K AE D IY”.

2.3. Incorporating OOV words into language model

To incorporate recovered OOV words into the recognizer’s LM, we need to estimate n -gram LM scores for these words. It is difficult to estimate LM scores for an OOV word, as we do not have any OOV text data except the decoded sentences in which the OOV word appears. We therefore use the syntactic and semantic properties of an OOV word.

In the hybrid decoding result, we estimate the POS label for OOV words using the Stanford MaxEnt POS tagger [25]. We adopt all 35 labels from the Penn Treebank POS tag set [26]. Then LM scores of an OOV word can be estimated from IV words in the same syntactic category - IV words with the same POS label. Precisely, the unigram score of OOV word w_i with POS label l_i at the i -th position in a sentence is calculated as

$$p(w_i) = \sum_{l_i} \frac{p(w_i|l_i)p(l_i)}{p(l_i|w_i)}, \quad (1)$$

where $p(l_i)$ is the prior probability of POS label l_i , $p(w_i|l_i)$ is the likelihood of observing OOV word w_i from all words with POS label l_i , while $p(l_i|w_i)$ is the probability of OOV word w_i having the POS label l_i . We sum over all l_i , because an OOV word may be labeled with different POS tags. In this paper, we obtain $p(l_i)$ from a POS label LM trained from the training text data. We approximate $p(w_i|l_i)$ as

$$p(w_i|l_i) = \frac{1}{N}, \quad (2)$$

where N is the number of IV words with POS label l_i in the training data. Furthermore, we calculate

$$p(l_i|w_i) = \frac{C(l_i, w_i)}{C(w_i)}, \quad (3)$$

where $C(w_i)$ is the count of OOV word w_i in the development data and $C(l_i, w_i)$ is the count of OOV word w_i with POS label l_i . Similarly, we can estimate the bigram and trigram LM score for OOV word w_i as

$$p(w_i|w_{i-1}) = \sum_{l_i} \frac{p(w_i|l_i)p(l_i|l_{i-1})}{p(l_i|w_i)}, \quad (4)$$

and

$$p(w_i|w_{i-1}, w_{i-2}) = \sum_{l_i} \frac{p(w_i|l_i)p(l_i|l_{i-1}, l_{i-2})}{p(l_i|w_i)}, \quad (5)$$

where l_{i-1} and l_{i-2} is the POS label of the word at the $(i-1)$ -th and $(i-2)$ -th position respectively. During our experiment, we find that $p(w_i|l_i)$ can be very small, especially for nouns, where N is very large. As a result, estimated LM scores of OOV words are usually much smaller than LM scores of IV words. We therefore set a floor on $p(w_i|l_i)$ to prevent it from being too small. The threshold is tuned on the development data to produce the best performance.

Using the above method, we can estimate LM scores for an OOV word in observed contexts - sentences containing the OOV

word. However, an OOV word may come up in an unseen context in future. Therefore we also try to estimate possible contexts an OOV word may appear by using its semantic properties collected from the WordNet [27]. Particularly, for an OOV word, we identify IV words that are similar to it by measuring their semantic relatedness using the information content of concepts in WordNet [28, 29]. We then explore two approaches to predict new contexts for an OOV word. In one method, we use n -grams belonging to IV words which are similar to the OOV word to build new OOV n -grams. For example, for recovered OOV word “APTITUDE”, we find similar IV words, such as “TALENT”, from WordNet. Then, for n -grams containing “TALENT”, we replace “TALENT” with “APTITUDE” to build new OOV n -grams. In another method, we use IV words that are similar to IV words surrounding an OOV word to build new n -grams. For example, “TEST” is the word after the OOV word “APTITUDE” in the recognition hypothesis. We then use similar words, such as “EXAM”, to replace “TEST” to build new n -grams for OOV word “APTITUDE”. However, from the experimental results on the development data, we find that it is very difficult to correctly predict new contexts for an OOV word. Furthermore, because we add many irrelevant n -grams into the LM, the recognition accuracy can be worse. Therefore, in the following experiments, we do not predict new OOV n -grams, but use learned OOV unigrams to recognize OOV words appearing in unseen contexts. In our future work, similar to [30], we will investigate the use of the large amount of data on the Internet to estimate new contexts for OOV words.

3. Experiment setup

3.1. The word-fragment hybrid system

We build word-syllable based hybrid systems from the Wall Street Journal (WSJ), Broadcast News (BN), and Switchboard (SWB) corpora, respectively. To have enough OOV words for training and testing and to maintain a comparable OOV rate to practical systems, we select the top 20k words as vocabulary for the WSJ and BN system, and the top 10k words for the SWB system. For WSJ, the development and testing data are selected from the WSJ '92 and '93 Eval sets; for BN, the 1996 HUB4 Eval data is used; for SWB, we test on a subset of the SWB2 data. Table 2 presents some statistics on the development and testing data of each task. We can find that these datasets have about 2% OOV words in the development and testing sets. We also notice that some OOV words in the testing data are repeating OOV words that already appear in the development set. In this paper, we perform OOV word detection and recovery on the development set to learn OOV words. Then we evaluate how many learned OOV words can be recognized when they appear again in the testing data.

Table 2: Statistics on the development and testing data.

Task	WSJ	BN	SWB
Development OOV Rate	2.2%	2.0%	1.7%
Testing OOV Rate	2.1%	2.8%	1.8%
OOV words in Testing that also in Development	52.3%	23.9%	66.1%

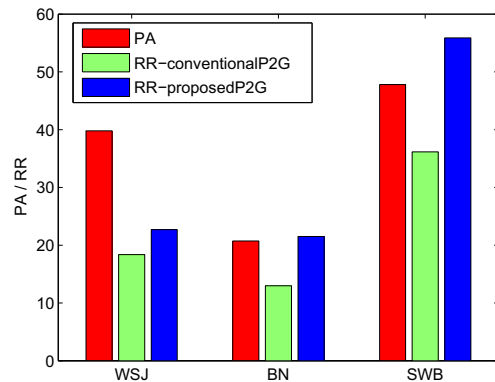


Figure 2: The OOV word recovery performance on the development data.

3.2. Evaluation metrics

We use precision and recall to measure the OOV word detection performance.

$$Precision = \frac{\#OOVs \text{ detected}}{\#OOVs \text{ reported}} \times 100\% \quad (6)$$

$$Recall = \frac{\#OOVs \text{ detected}}{\#OOVs \text{ in reference}} \times 100\% \quad (7)$$

We use pronunciation accuracy (PA) and recovery rate (RR) to measure the OOV word recovery performance.

$$PA = \frac{\#OOVs \text{ detected with correct pronunciation}}{\#OOVs \text{ detected}} \times 100\% \quad (8)$$

$$RR = \frac{\#OOVs \text{ recovered with correct spelling}}{\#OOVs \text{ detected}} \times 100\% \quad (9)$$

Finally, we calculate the word error rate (WER) to evaluate the overall performance of our speech recognition system.

4. Experiment results

4.1. The OOV word detection performance

From the OOV word detection performance in Table 3, we find that the hybrid system performs very well in the WSJ and SWB tasks - precision is more than 60% and up to 75% OOV words are detected. But in the BN task, utterances are usually much longer than those in the WSJ and SWB tasks and multiple OOV words can appear in one utterance or even in a sequence, which make OOV word detection more difficult.

Table 3: The OOV word detection performance on the development data.

Task	WSJ	BN	SWB
Precision	63.8%	49.8%	67.2%
Recall	74.0%	62.4%	74.6%

4.2. The OOV word recovery performance

After detecting OOV words in the development data, we try to recover their written form. The OOV word recovery performance on the development data is shown in Figure 2. It can

be seen that less than 50% OOV words are decoded with the correct pronunciation. If we perform the P2G conversion using the conventional model as described in [18], even fewer OOV words are recovered with the correct orthography. Instead, when using the proposed P2G model, we are able to correctly estimate the written form for more OOV words. In the BN and SWB tasks, RR is even larger than PA, as many OOV words are now recovered from incorrect pronunciations. The improvement in the WSJ task is smaller than that in the BN and SWB tasks. This may be because we have fewer negative examples in the WSJ task when training the proposed P2G model.

Having the pronunciation and written form of an OOV word, we can integrate recovered OOV words into our recognizer's lexicon. Table 4 provides the size and OOV rate of the OOV expanded lexicon. Comparing with Table 2, we can find that by only increasing the vocabulary size about 1%, the OOV rate of both the development and testing data is significantly reduced. On average, more than 40% OOV words are successfully learned from the development data data.

Table 4: The size and OOV rate of the OOV expanded lexicon.

Task	WSJ	BN	SWB
Vocabulary Size	20267	20305	10145
Development OOV Rate	1.2%	1.5%	0.9%
Testing OOV Rate	1.6%	2.5%	1.2%

4.3. The recognition result on the testing data

Before evaluating on the testing data, we need to incorporate learned OOV words into the recognizer's LM. We first perform POS tagging on all words in the hybrid decoding result, where the POS tagging accuracy on OOV words is about 80%. Then we estimate LM scores for OOV words using POS labels of the word and its surrounding words. The number of new n -grams added into the recognizer's LM is provided in Table 5. It can be seen that only a small number of new n -grams are added.

Table 5: The number of n -grams added into the recognizer's LM.

Task	WSJ	BN	SWB
Unigram	267	305	145
Bigram	536	621	312
Trigram	683	860	442

The goal of learning OOV words is to correctly recognize them if encountered by the recognizer in future. Therefore we count how many repeating OOV words – OOV words that appear in both development and testing data, can be successfully learned from the development data and then correctly recognized in the testing data. Figure 3 shows the percentage of learned repeating OOV words that are recognized in the testing data when gradually expanding the LM with OOV unigrams, bigrams and trigrams. The red bar is the result when only adding unigram scores, the green bar corresponds to adding both unigram and bigram scores, and the blue bar is the result when adding all OOV n -grams. We can find that by only expanding the LM with OOV unigrams, we can already recognize more than 50% learned repeating OOV words. Because many repeating OOV words appear in unseen contexts in the testing data, adding bigram and trigram scores into the LM does not help

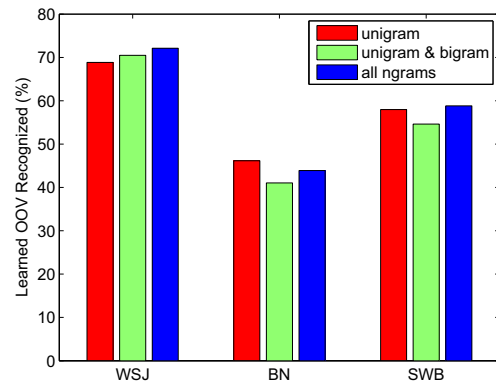


Figure 3: The percentage of learned repeating OOV words are recognized in the testing data.

very much. On average, we can recognize about 60% learned repeating OOV words in the testing data.

Finally, let us examine how the overall WER changes after expanding the recognizer's vocabulary with learned OOV words. From Table 6, it can be seen that because we now correctly recognize about 60% learned repeating OOV words, the WER on the testing data is improved. In Table 6, we also present the approximate lower bound of the overall WER by assuming that we can correctly recognize all repeating OOV words in the testing data. We can see that our recognition performance is very close to the approximate lower bound. The true lower bound should be slightly smaller than the approximate one, as we should make fewer recognition errors on surrounding IV words of correctly recognized OOV words.

Table 6: The WER on the testing data when decoding with different LMs.

Task	WSJ	BN	SWB
No OOV Learning	10.1%	30.4%	33.3%
Unigram	9.4%	30.2%	32.7%
Unigram & Bigram	9.3%	30.2%	32.7%
All n -grams	9.3%	30.2%	32.6%
Approximate Lower Bound	9.0%	29.7%	32.1%

5. Conclusion

This paper introduces a learning scheme for speech recognition system that can automatically expand its vocabulary with new words inferred from OOV regions in testing speech. We describe how to detect and recover OOV words and then integrate them into the recognizer's lexicon. We also propose a method to add OOV words into the recognizer's LM by using their syntactic and semantic properties. From the experimental results, we find that more than 40% OOV words are successfully learned from the development data and about 60% learned OOV words are correctly recognized in the testing data. Furthermore, by recognizing learned repeating OOV words, the overall WER is improved. Because many OOV words may appear in unseen contexts when encountered by the recognizer in future, our next step is to investigate the use of extensive data from Web sources to predict new contexts for OOV words.

6. References

- [1] S. Hayamizu, K. Itou, and K. Tanaka, "Detection of unknown words in large vocabulary speech recognition," *Proc. Eurospeech*, pp. 2113-2116, 1993.
- [2] H. Lin, J. Bilmes, D. Vergyri, and K. Kirchhoff, "OOV detection by joint word/phone lattice alignment," *Proc. ASRU-2007*, pp. 478-483, 2007.
- [3] L. Burget, P. Schwarz, P. Matejka, H. Hermansky, and J. Cernocky, "Combining of strongly and weakly constrained recognizers for reliable detection of OOVs," *Proc. ICASSP-2008*, pp. 4081-4084, 2008.
- [4] C. White, G. Zweig, L. Burget, P. Schwarz, and H. Hermansky, "Confidence estimation, OOV detection and language ID using phone-to-word transduction and phone-level alignments," *Proc. ICASSP-2008*, pp. 4085-4088, 2008.
- [5] S. Kombrink, L. Burget, P. Matejka, M. Karafiat, and H. Hermansky, "Posterior-based out-of-vocabulary word detection in telephone speech," *Proc. Interspeech-2009*, pp. 80-83, 2009.
- [6] H. Sun, G. Zhang, F. Zheng, and M. Xu, "Using word confidence measure for OOV words detection in a spontaneous spoken dialog system," *Proc. Eurospeech-2003*, pp. 2713-2716, 2003.
- [7] B. Lecouteux, G. Linares, B. Favre, "Combined low level and high level features for out-of-vocabulary word detection," *Proc. Interspeech-2009*, pp. 1187-1190, 2009.
- [8] F. Stouten, D. Fohr, and I. Illina, "Detection of OOV words by combining acoustic confidence measures with linguistic features," *Proc. ASRU-2009*, pp. 371-375, 2009.
- [9] D. Klakow, G. Rose, and X. Aubert, "OOV-detection in large vocabulary system using automatically defined word-fragments as fillers," *Proc. Eurospeech-1999*, pp. 49-52, 1999.
- [10] I. Bazzi and J. Glass, "Modeling out-of-vocabulary words for robust speech recognition," *Proc. ICSLP-2000*, vol. 1, pp. 401-404, 2000.
- [11] T. Schaaf, "Detection of OOV words using generalized word models and a semantic class language model," *Proc. Eurospeech-2001*, pp. 2581-2584, 2001.
- [12] L. Galescu, "Recognition of out-of-vocabulary words with sub-lexical language models," *Proc. Eurospeech-2003*, pp. 249-252, 2003.
- [13] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," *Proc. Interspeech-2005*, pp. 725-728, 2005.
- [14] A. Rastrow, A. Sethy, B. Ramabhadran, and F. Jelinek, "Towards using hybrid, word and fragment units for vocabulary independent LVCSR systems," *Proc. Interspeech-2009*, pp. 1931-1934, 2009.
- [15] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, "Contextual information improves OOV detection in speech," *Proc. HLT-NAACL-2010*, pp. 216-224, 2010.
- [16] M. Shaik, A. El-Desoky, R. Schluter, and H. Ney, "Hybrid language model using mixed types of sub-lexical units for open vocabulary German LVCSR," *Proc. Interspeech-2011*, pp. 1441-1444, 2011.
- [17] I. Bulyko, J. Herrero, C. Mihelich, and O. Kimball, "Sub-word speech recognition for detection of unseen words," *Proc. Interspeech-2012*, 2012.
- [18] L. Qin, M. Sun, and A. Rudnicky, "OOV detection and recovery using hybrid models with different fragments," *Proc. Interspeech-2011*, pp. 1913-1916, 2011.
- [19] L. Qin, M. Sun, and A. Rudnicky, "System combination for out-of-vocabulary word detection," *Proc. ICASSP-2012*, pp. 4817-4820, 2012.
- [20] L. Qin and A. Rudnicky, "OOV word detection using hybrid models with mixed types of fragments," *Proc. Interspeech-2012*, 2012.
- [21] L. Qin and A. Rudnicky, "Finding recurrent out-of-vocabulary words," *Proc. Interspeech-2013*, 2013.
- [22] L. Qin and A. Rudnicky, "Learning better lexical properties for recurrent OOV words," *Proc. ASRU-2013*, 2013.
- [23] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434-451, 2008.
- [24] S. F. Chen, "Conditional and joint models of grapheme-to-phoneme conversion," *Proc. Eurospeech-2003*, pp. 2033-2036, 2003.
- [25] K. Toutanova, D. Klein, C. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," *Proc. HLT-NAACL-2003*, pp. 252-259, 2003.
- [26] M. P. Marcus, M. Marcinkiewicz and B. Santorini, "Building a large annotated corpus of English: the penn treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313-330, 1993.
- [27] G. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, Vol. 38, No. 11, pp. 39-41, 1995.
- [28] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity: measuring the relatedness of concepts," *Proc. HLT-NAACL-2004*, pp. 38-41, 2004.
- [29] Lin D, "An information-theoretic definition of similarity," *Proc. ICML-1998*, pp. 296-304, 1998.
- [30] A. Gandhe, L. Qin, F. Metze, A. Rudnicky, I. Lane, and M. Eck, "Using web text to improve keyword spotting in speech," *Proc. ASRU-2013*, 2013.