*INTERSPEECH 2016*
September 8–12, 2016, San Francisco, USA

# Web Data Selection Based on Word Embedding for Low-Resource Speech Recognition

*Chuandong Xie[1], Wu Guo[1], Guoping Hu[2], Junhua Liu[3]*

[1] National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, China
[2] Key Laboratory of Intelligent Speech Technology, Ministry of Public Security, Hefei, P.R.China
[3]iFLYTEK Research, Hefei, China

xcdahu@mail.ustc.edu.cn, guowu@ustc.edu.cn, gphu@iflytek.com,jhliu@iflytek.com

## Abstract

The lack of transcription files will lead to a high out-of-vocabulary (OOV) rate and a weak language model in low-resource speech recognition systems. This paper presents a web data selection method to augment these systems. After mapping all the vocabularies or short sentences to vectors in a low-dimensional space through a word embedding technique, the similarities between the web data and the small pool of training transcriptions are calculated. Then, the web data with high similarity are selected to expand the pronunciation lexicon or language model. Experiments are conducted on the NIST Open KWS15 Swahili VLLP recognition task. Compared with the baseline system, our methods can achieve a 5.23% absolute reduction in word error rate (WER) using the expanded pronunciation lexicon and a 9.54% absolute WER reduction using both the expanded lexicon and language model.

**Index Terms**: word embedding, web data, lexicon, language model

## 1. Introduction

In recent years, there has been growing interest in large vocabulary continuous speech recognition (LVCSR) and keyword spotting (KWS) for low-resource languages. Three key components are required in LVCSR: an acoustic model (AM), a language model (LM) and a pronunciation lexicon. Only a very small amount of transcribed speech can be used in low-resource LVCSR, resulting in a small lexicon with high OOV rate and a weak acoustic and language model. A direct solution is to utilize out-of-domain data to solve these problems. In this paper, we focus on using web data to augment the lexicon and the language model.

Much prior work has proposed to collect web data to improve the recognition accuracy. To obtain useful web data, [1] summarized five main steps to scrape large amounts of conversational data and [2] elaborated Rapid Language Adaptation Toolkit (RLAT) with RSS (really simple syndication) Feeds-based crawling methods for collecting large amounts of web data. Most web data are different from the conversational transcripts in format, and methods were introduced to pre-process and normalize the collected web data in [3]. After pre-processing, different criteria are used to select web data. [4, 5] adopted the relative entropy based query generation mechanism to download documents, and [6] used perplexity as a similarity measure between in-domain data and web data. How to combine the web data and original transcriptions in LM training is also an important research topic. R. Iyer *et al*. proposed to combine out-of-domain data with domain dependent data to improve statistical LM performance [7]. An efficient query selection algorithm for the retrieval of web text data to augment a statistical language model was presented in [8], and [9] adjusted counts of N-grams from querying the web page and interpolated them with traditional corpus-based tri-gram estimates. Despite the aforementioned works, it is still a difficult problem to determine a criterion to match the web data with conversational speech in similar style and topic.

Recently, Mikolov *et al*. proposed word embedding technique for continuous word representations in vector space [10, 11, 12, 13]. This has been widely used in various natural language processing tasks, including neural language model [14], machine translation [15] and antonym selection [16]. Various models, such as the Skip-gram model [10, 11], have been proposed to take advantage of the context of each word in large corpora to learn word embedding. The word2vec toolkit [17] can be downloaded from web site to train Skip-gram models. In [18], the authors adopted the Skip-gram model architecture to capture the lexical and semantic relations to retrieve OOV proper names.

In this paper, we first use a Skip-gram model to map all the words to a continuous vector space. Then, the similarity scores of words between the training set and the web data are calculated; words with similarity score above the threshold are selected to expand the decoding lexicon. Furthermore, the K-means algorithm is used to cluster the training transcripts into a small number of topics. The similarity scores between these clustering topics and the web data sentences are also calculated, and the sentences with high scores are selected to augment the language model.

The remainder of this paper is organized as follows. We present the dataset used for our evaluations in Section 2. Section 3 elaborates the proposed methodology. Section 4 describes the experimental setup. Experimental results are presented in Section 5. Concluding remarks are provided in Section 6

## 2. Dataset

We conduct our investigations on the NIST OpenKWS15 Swahili language recognition task, and the IARPA Babel Program language collection IARPA-Babel {202b-v1.0d Swahili} very limited language packs (VLLP) corpus is used



http://dx.doi.org/10.21437/Interspeech.2016-45

in the experiments. The training set of Swahili VLLP consists of 3 hours of transcribed speech, and the lexicon includes only 4,957 words. The 10-hour development set is used to evaluate the speech recognition performance in this paper. Furthermore, two web data sources, provided by BBN and IBM, are used to fulfil the active learning task. The details of the training set and the web data are listed in Table 1. Compared with the web data, the 3-hour transcriptions of the training set contain much fewer words and sentences. The BBN web data size is much larger than that of IBM web data.

*Table 1: The details of the supplied data*

| Data Sources | Number of words | Data size ( MB) |
|---|---|---|
| Training set | 4,957 | 0.16 |
| IBM | 46,048 | 10.6 |
| BBN | 379,434 | 263 |

# 3. Proposed methodology

## 3.1. Web Data Preprocessing

The web data contains substantial garbage text and cannot be directly used in automatic speech recognition (ASR). We carry out the following steps to preprocess the web data:

(1) The html format files are converted to plain text format file. The internet addresses and headers are first stripped. Non-standard whitespaces are replaced with a standard version, and special symbols are removed. Some of the upper-case letters are converted to lower-case. To use the web data text effectively for lexicon expanding and language modelling, some special abbreviations are converted to the mapping format (e.g., "imeng'ara" to "imengQara"); the digit sequences are normalized to correspond to their Swahili natural spoken form (e.g., "101" to "mia moja na moja"); and other "non-standard words" (e.g., "SMZ", "CBG") are removed. The sentences are segmented based on some special punctuation, such as semicolons and exclamatory marks. The original web data are saved in a number of small files, and we merge all these small files into a large plain-text file.

(2) Non-Latin characters are removed. The provided web data are mixed with characters from many other languages, which must be removed. Because the web data are saved in a plain text format, all the sentences can be arranged in alphabetical order using the Linux "sort" command. The Swahili adopts the Latin character set, and we can remove the sentences that contain any non-Latin characters.

(3) Non-Swahili Latin characters are removed. Except for a few characters, the character set of the Swahili language is also used in other Latin languages. The web data after step 2 still contains many non-Swahili words, especially some English words. In this step, we remove the words that include letters not in the character set of the Swahili language.

## 3.2. The Skip-gram model

Because the vocabulary size of web data is much larger than that of conversational speech, a criterion must be found to select web data. In this paper, the Skip-gram model is utilized to learn word embedding. The goal of the Skip-gram model is to learn word representations that can predict the surrounding words or phrases in a sentence or some documents. Each word in corpora is mapped to a continuous embedding space by looking up an embedding matrix $V^{(1)}$, which is learned by maximizing the prediction probability that is calculated from its neighboring words within a context window, and the prediction matrix is $V^{(2)}$.

Given a sequence of $T$ training words, denoted by $w_1, w_2, w_3, \cdots, w_T$, the goal of the Skip-gram model is to maximize the following average log probability:

$$G = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\ne 0} \log p(w_{t+j} \mid w_t) \tag{1}$$

where $C$ is the context window size, $w_t$ is the input central word and $w_{t+j}$ is neighboring words of $w_t$. The Skip-gram model defines the above conditional probability $p(w_{t+j} \mid w_t)$ using the following soft-max function:

$$p(w_{t+j} \mid w_t) = \frac{\exp(v_{t+j}^{(2)} \cdot v_t^{(1)})}{\sum_{k=1}^{T} \exp(v_k^{(2)} \cdot v_t^{(1)})} \tag{2}$$

where $v_t^{(1)}$, $v_k^{(2)}$ are from row vector representations corresponding to word $w_t$, $w_k$ in matrices $V^{(1)}$, $V^{(2)}$ respectively.

The process of training the Skip-gram model can be described as an optimization problem to maximize the above objective function $G$, and the optimization problem can be solved by the stochastic gradient descent (SGD) method [12].The embedding matrix $V^{(1)}$ can be learned by the Skip-gram model and used as the word embedding for all words in the data sets.

## 3.3. Web data selection for lexicon expansion

In this paper, we propose to select the similar words or semantic correlation words with the training set from web data to expand the decoding lexicon through word embedding method. Figure 1 shows the flowchart for expanding the decoding lexicon using the web data.

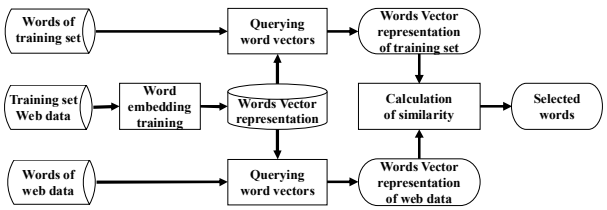


*Figure 1. Using word embedding to expand the decoding lexicon*

All the data, including the transcriptions of the training set and web data, are first used to learn word embedding by the Skip-gram model. Then, all the words are converted into vectors through word embedding. Cosine similarity can be calculated between the words of the training set and the web

data. Given two vectors $\vec{X} = (x_1, x_2, \cdots, x_n)$ and $\vec{Y} = (y_1, y_2, \cdots, y_n)$, the cosine score is as follows:

$$sim(\vec{X}, \vec{Y}) = \frac{\vec{X} \cdot \vec{Y}}{\| \vec{X} \| \cdot \| \vec{Y} \|} \quad (3)$$

For each word in the training set, only the top-*N* words from web data with the highest cosine score are chosen to expand the decoding lexicon. Different *N* values can significantly influence the performance of speech recognition; our later experiments demonstrate this viewpoint.

### 3.4. Web data selection for language model

In this section, we use word embedding and K-means method to select web data to improve the language model for speech recognition. The framework of the proposed method is illustrated in Figure 2.

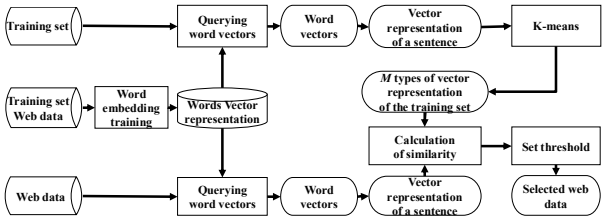


*Figure 2. Using word embedding and K-means method to select web data.*

Similar to Figure 1, all the data from the web data and the training set are first used to learn word embedding using the Skip-gram model. To obtain the vector representation of one sentence, all the word vectors of this sentence are first summed and then divided by the number of words. After changing the sentences of the training set into vectors, the K-means algorithm is used to partition all these vectors into *M* clusters.

In the data selection procedure, the sentences of web data are also converted into vectors. These vectors are used to calculate the similarity with the afore-mentioned *M* clusters of the training set. Cosine score is also used as a similarity measurement. After calculating the *M* similarity scores, a threshold is set to select the sentences. In our view, the selected sentences from the web data are matching with the training set in style or topic.

The web data selected by the proposed method can be used to train the language model. The state-of-the-art algorithm for using additional web data is to train separate LMs for the different sources using a unified vocabulary and then combine them by interpolation. This method is also referred to as mixtures of LMs. We adopt this method in this paper. The interpolation weights are tuned in the tuning set provided by NIST.

## 4. Experimental Setup

For our experiments, Kaldi tools [19] are adopted to train the Deep Neural Networks and Hidden Markov Model (DNN-HMM) [20]. Morfessor toolkit [21] is used to segment words into morphemes, since pronunciation lexicons are not provided for the Babel VLLPs.

The 43-dimensional Mel Frequency Cepstral Coefficients (MFCC) and pitch features are used in the experiments. Each speech signal is parameterized by the 13th order MFCCs and their first and second derivatives, and 4-dimension pitch features [22], forming a 43-dimension feature vector. For DNN training, multi-task training [23] is used to initialize the DNN model, and the other language packs provided by NIST OpenKWS 2015 [24] are used in multi-task training. Furthermore, 20 hours of unsupervised data are added to the training set for acoustic modelling.

We compare the effect of different lexicons and LMs in the ASR experiments. The lexicons are formed using the original training transcriptions and selected web data. Word based trigram LMs are used in decoding, and the LMs are trained with Kneser-Ney smoothing using the SRI LM toolkit [25].

In our experiments, the word2vec toolkit [17] is used to learn word embedding. The size of context window is set to 5, the dimension of the learned vectors is set to 50, and the lowest frequency of words is set to 1.

## 5. Experiments & Results

### 5.1. Results of lexicon expansion

Only the transcription of 3-hours training set is used to train the acoustic model and the LM in the baseline system, and the word size of the lexicon is 4,957. Table 2 shows the ASR word error rate (WER) of the baseline on the 10 hours VLLP development set.

*Table 2: The baseline results on the VLLP development set*

| Number of lexicon words | Word Error Rate (WER)/% |
|---|---|
| 4,957 | 61.45 |

In a subsequent experiment, we evaluate the effectiveness of web data in expanding the decoding lexicon for ASR. We select the similar words or semantic correlation words with the training set from web data to expand the decoding lexicon. As mentioned in 3.3, only the top-*N* words from web data with the highest cosine score are chosen to expand the decoding lexicon. Table 3 presents the results of ASR WER with different sizes of selected words from web data. The LMs are trained on the 3-hour training set.

*Table 3: The results of word embedding method to select the similar words with training set from web data to expand the decode lexicon*

| Number of web data words selected | Number of lexicon words | WER/% |
|---|---|---|
| 27,847 | 32,804 | 56.25 |
| 35,540 | 40,497 | 56.22 |
| 52,517 | 57,474 | 57.01 |
| all | 430,439 | 58.48 |

Compared with the baseline system in Table 2, all the systems with expanded lexicons can improve the ASR performance. The best performance with 40,497 lexicon words can yield an absolute 5.23% WER reduction. In the last row of Table 3, all the words in the web data are used in the decoding

lexicon, and an absolute 2.97% WER reduction can still be observed. The best system in row 2 can outperform the system in the last row by an absolute 2.26% WER reduction. One possible explanation is that word embedding method can efficiently select similar words or semantic correlation words with the training set.

### 5.2. Results of web data selection for language model

For validating the web data in LM modelling, we first conduct ASR experiments based on different LMs. The first LM (LM1) is trained on only the training set. The second LM (LM2) is a linear interpolation language model of different LMs that are separately trained on the training set and the web data. In these experiments, no word embedding data selection is applied and all the available web data are used to build the LM. The interpolation weights for each sub-LM and the final best results of speech recognition with different decoding lexicon are presented in Table 4.

*Table 4: The best speech recognition results of the training set and web data mixture language models*

| Number of lexicon words | Language Sources | LM Weights | Word Error Rate (WER)/% |
|---|---|---|---|
| 40,497 | VLLP | 1.0 | 56.22 |
| 40,497 | VLLP<br>IBM<br>BBN | 0.848<br>0.128<br>0.024 | 54.08 |
| 430,439 | VLLP | 1.0 | 58.48 |
| 430,439 | VLLP<br>IBM<br>BBN | 0.60<br>0.25<br>0.15 | 55.42 |

As can be seen in Table 4, LM2 trained with additional web data can achieve better ASR performance than LM1. In Table 4, LM2 with only 40,497-word lexicon can yield the best result, and it can obtain a 1.34% absolute WER reduction against LM2 with all words from the web data (430,439). For the linear interpolation weights of LM2, the sub-LM trained with VLLP transcription has the largest weight, because the development set matches the training set best. BBN web data have the smallest weight; possibly because its style and topic are different from the development set.

*Table 5: The result of using perplexity based approach to improve the language model*

| Number of lexicon words | Language Sources | LM Weights | WER/% |
|---|---|---|---|
| 40,497 | VLLP<br>IBM_PPL_selected<br>BBN_PPL_selected | 0.887<br>0.068<br>0.045 | 52.76 |

For comparison with our methods, a perplexity (PPL) [26, 27] based approach to select web data is also implemented in our experiments. In this experiment, the value of perplexity is set to 4000. Table 5 shows the best performance of using perplexity based approach to improve the language model with lexicon 40497. Compared with the best result in Table 4, the linearly interpolated LM2 with selected web data by perplexity approach can obtain a 1.32% absolute improvement.

To evaluate the effectiveness of the proposed web data selection method, we implement a confirmatory experiment based on the best result above. Table 6 illustrates the benefit of using word embedding method to select web data in LM modelling. In this experiment, the number of clusters $M$ is set to 5.

*Table 6: The result of using word embedding and K-means method to improve the language model*

| Number of lexicon words | Language Sources | LM Weights | Word Error Rate (WER)/% |
|---|---|---|---|
| 40,497 | VLLP<br>IBM_selected<br>BBN_selected | 0.848<br>0.128<br>0.024 | 51.91 |

Compared with the best result in Table 4, the linearly interpolated LM2 with selected web data can obtain a 2.17% absolute improvement. Compared with the LM1 in Table 4, it can yield 4.31% absolute improvement. Compared with the baseline system in Table 2, this method can achieve a 9.54% absolute WER reduction. Contrasted with the best result listed in Table 5, our proposed web data selection method performs better, 0.85% absolute WER reduction, than perplexity approach.

From Table 4 and Table 6, we observe that the selected web data in LM modelling can improve ASR performance. The main reason for this improvement is that the word embedding method can find the sentences similar to the training set in style or topic, and filter out the data that are not reliable or relevant to the training set.

## 6. Conclusions

Word embedding with good semantic representations is invaluable to many natural language processing tasks. In this paper, we have investigated how to apply this method to process web data for low-resource languages ASR. First, we apply a word embedding method to select similar words or semantic correlation words with training set from web data to expand the decoding lexicon. Then, we use word embedding and K-means methods for sentence selection to improve the language model. The experimental results demonstrate that our proposed methods can significantly improve ASR performance. Thus, the word embedding method can be used effectively to select words or data similar to the training set in style or topic.

## 7. Acknowledgements

This work was partially funded by National Key Technology Support Program (Grant No. 2014BAK15B05) and the Natural Science Foundation of Anhui Province (Grant No. 1408085MKL78).

## 8. References

[1] Gideon Mendels, Erica Cooper, Victor Soto, Julia Hirschberg, "Improving Speech Recognition and Keyword Search for Low Resource Languages Using Web Data," *in Proceedings of Interspeech2015,* pp.829-833, 2015.

[2] T. Schlippe, L. Gren, N. T. Vu, and T. Schultz, "Unsupervised language model adaptation for automatic speech recognition of broadcast news using web 2.0," *in Proceedings of Interspeech*, pp. 2698–2702, 2013.

[3] I. Bulyko, M. Ostendorf, M. Siu, T. Ng, A. Stolcke, and O. C¸ etin, "Web resources for language modeling in conversational speech recognition," *ACM Trans. Speech Lang. Process.* , vol. 5, no. 1, pp. 1:1–1:25, Dec. 2007.

[4] Abhinav Sethy, Bhuvana Ramabhadran, and Shrikanth Narayanan, "Measuring convergence in language model estimation using relative entropy," *in Proceedings of ICSLP*, 2004.

[5] A. Sethy, P. G. Georgiou and S. Narayanan, "Building topic specific language models from web data using competitive models," *in Proceedings of Interspeech2005*, pp. 1293-1296, 2005.

[6] Ankur Gandhe, Long Qin, Florian Metze, Alexander Rudnicky, Ian Lane, Matthias Eck ,"Using web text to improve keyword spotting in speech," *IEEE Workshop on Automatic Speech Recognition & Understanding,* pp. 428-433, 2013.

[7] R. Iyer, M. Ostendorf, and H. Gish, "Using out-of-domain data to improve in-domain language models," *in IEEE Signal Processing Letters,* no. 8, pp. 221-223, 1997.

[8] M. Creutz, S. Virpioja, and A. Kovaleva, "Web augmentation of language models for continuous speech recognition of SMS text messages," *in Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009). Athens, Greece: Association for Computational Linguistics*, pp. 157–165, March 2009.

[9] X. Zhu and R. Rosenfeld, "Improving trigram language modeling with the World Wide Web," *in Proc. ICASSP, 2001*, pp.I:533-536, 2001.

[10] Mikolov, T., Chen, K., Corrado, G. and Dean, J. "Efficient Estimation of Word Representations in Vector Space," *Proceedings of Workshop at ICLR*, 2013.

[11] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. "Distributed Representations of Words and Phrases and their Compositionality," *Proceedings of NIPS*, 2013.

[12] Mikolov, T., Yih, W. and Zweig, G. "Linguistic Regularities in Continuous Space Word Representations," *in Proceedings of NAACL HLT*, 2013.

[13] Geoffrey E Hinton, James L McClelland, and David E Rumelhart. Distributed representations. In Parallel distributed processing: Explorations in the microstructure of cognition. *MIT Press, Volume 1: Foundations*, pp. 77-109, 1986.

[14] Holger Schwenk. "Continuous space language models." *Computer Speech & Language,* 21(3):492–518, 2007.

[15] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. "Fast and robust neural network joint models for statistical machine translation," *In Proceedings of ACL*, pp. 1370-1380, 2014.

[16] Zhigang Chen, Wei Lin, Qian Chen, Xiaoping Chen, Si Wei, Xiaodan Zhu, and Hui Jiang. "Revisiting word embedding for contrasting meaning," *In Proceedings of ACL,* pp. 106-112, 2015.

[17] Google, https://code.google.com/p/word2vec/.

[18] Dominique Fohr,Irina Illina, "Continuous Word Representation using Neural Networks for Proper Name Retrieval from Diachronic Documents", *Interspeech2015*, pp.3506-3510, 2015.

[19] D.Povey,A.Ghoshal,G.Boulianne,L.Burget,O.Glembek, N. Goel, M. Hannermann, P. Motlícek, Y. Qian, P. Schwartz, J. Silovský, G. Stemmer, and K. Veselý, "The kaldi speech recognition toolkit," *in ASRU . IEEE,* 2011.

[20] Geoffrey Hinton, Li Deng, Dong Yu, et. al, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, pp. 82-97, November 2012

[21] Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, "Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline," *Aalto University Publication,* 2013.

[22] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," *in ICASSP .IEEE,* pp. 2494-2498, 2014.

[23] Z. Wu, C. Botinhao, O. Watts, and S. King. "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," *in Proceedings of ICASSP*, pp. 4460-4464, 2015.

[24] NIST, http://www.nist.gov/itl/iad/mig/upload/KWS15-evalplan-v05.pdf.

[25] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit," *in Proceedings of ICSLP,* pp. 901-904, 2002.

[26] Helin Dutağacı, "Statistical language models for large vocabulary Turkish speech recognition," *B.S. in E.E., Boğaziçi University,* 1999

[27] Metze F, Gandhe A, Miao Y, et al. "Semi-supervised training in low-resource ASR and KWS," *in Proceedings of ICASSP* 2015.