INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Class-Based Neural Network Language Model for Second-Pass Rescoring in ASR

Lingfeng Dai[#], *Qi Liu*[#], *Kai Yu*[*]

MoE Key Lab of Artificial Intelligence
X-LANCE Lab, Department of Computer Science and Engineering
AI Institute, Shanghai Jiao Tong University, Shanghai, China
{randool,liuq901,kai.yu}@sjtu.edu.cn

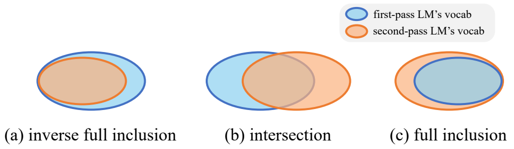
Abstract

Language model rescoring, especially neural network language model (NNLM) rescoring, is widely used to achieve improved performance in a second-pass automatic speech recognition (ASR) system. The rescoring NNLM is usually trained separately from the ASR system. Typically, the two's training corpora are different, leading to the vocabulary mismatch problem, consequently degrading ASR performance. Previous research focuses more on the language domain mismatch problem, while the vocabulary mismatch problem, which may also cause significant performance degradation, has not been well studied. This paper proposes a novel class-based NNLM framework to address the vocabulary mismatch problem for language model rescoring. Here, OOV words (unknown words to the rescoring NNLM are called OOV words for short) are assigned to well-trained classes of NNLM and inherit the class probability. Experiments show that class-based NNLM rescoring can significantly reduce performance degradation due to vocabulary mismatch.

Index Terms: neural network language model, second-pass rescoring, class-based softmax, vocabulary mismatch, parameter estimation

1. Introduction

The language model, calculating the probability of given sentences, is widely used in many fields, including machine translation, video captions, and speech recognition. Recently, recurrent neural network language models (NNLM) [1, 2], especially long short-term memory (LSTM) [3] language models, have achieved better results compared with the n-gram language model in the ASR system and hence become an indispensable part of the second-pass rescoring model in the ASR system. However, the applications of the rescoring NNLM lack research. Many studies have devoted to making the language model more accurate under the rescoring ASR framework, but few works focus on the vocabulary mismatch problem. Typically, NNLM is trained separately from the ASR system in the corpus, leading to a mismatch between the NNLM's and the ASR system's vocabulary. Figure 1 illustrates three possible relationships between the ASR system's vocabulary and the rescoring NNLM's, and the second relationship, which is most common, will cause the above phenomenon. Due to the vocabularies mismatch problem, the n-best sentences generated by the first-pass language model will contain OOV words, making NNLM inaccurately score these sentences [4]. In extreme cases, the rescoring result can be worse than the fundamental prediction. Furthermore, this problem cannot be solved simply by expanding the vocabulary due to the long-tail distribution of word frequency, which means if one wants to cover more OOV words, an additional exponential corpus is required.



(a) inverse full inclusion (b) intersection (c) full inclusion

Figure 1: *Three possible relationships between the vocabularies of first-pass LM and the second-pass rescoring LM.*

The commonly used method is to use a particular word, UNK, to represent all OOV words. However, this approach has some disadvantages [5]. First, all the OOV words are treated as the same UNK token, which is ambiguous. Second, each OOV word's frequency is accumulated to the UNK token, leading to abnormal probability.

One kind of method is using sub-word language models [6, 7, 8]. But the performance of sub-word language models is weaker than word-based language models. Another kind of work tries to address this problem using a class-based n-gram language model [9]. The OOV words are assigned to the proper class by semantics similarity. Morphological features are also used to cluster OOV words [10, 11]. [12, 13] use embedding augmentation and prediction. But these works cannot handle the OOV words encountered in the rescoring stage. [14, 15] use FOFE [16] and sub-word language model to extend the vocabulary of NNLM, but they need extra data contains particular OOV words to train the model.

This paper proposes a novel way to address the OOV words problem in NNLM rescoring. The main idea is: parameters of OOV words can be estimated by their synonyms. Specifically, the vocabulary of the rescoring NNLM is firstly clustered by the knowledge-driven method or the data-driven method. Second, the NNLM with a class-based softmax output layer is trained. Third, the proper class is assigned to each OOV word. Finally, OOV word's parameters, which are predicted by words in the corresponding class, can be used to score the sentences. The experiments show that our methods can significantly reduce the vocabulary mismatch problem during NNLM rescoring.

The rest of the paper is organized as follows. Section 2 briefly introduces the class-based NNLM. Section 3 shows the word clustering method of both knowledge-driven and data-driven. Section 4 demonstrates how to address the OOV problem with the class-based NNLM. Section 5 presents the experimental results and section 6 is the conclusion.

[#] equal contribution
[*] corresponding author

Copyright © 2021 ISCA

http://dx.doi.org/10.21437/Interspeech.2021-1080

2. Class-based NNLM

The language model models the probability of a sentence $W = w_1, w_2, \ldots, w_m$ by factorizing it into a series of the product of conditional probabilities

$$P(W) = \prod_{i=1}^{m} P(w_i | w_1, \ldots, w_{i-1}). \tag{1}$$

The NNLM with class-based softmax [17] (called class-based NNLM for short) calculates the conditional probability of each position i using the follow form:

$$\begin{aligned} P(w_i | W_{<i}) &= P(c_{w_i} | W_{<i}) P(w_i | c_{w_i}, W_{<i}) \\ &= P(c_{w_i} | h_i) P(w_i | c_{w_i}, h_i), \end{aligned} \tag{2}$$

where w_i denotes the i-th word in the sentence, and $W_{<i}$ represents w_i's history, which is $w_1, w_2, \ldots, w_{i-1}$. c_{w_i} is the category (or class) of w_i. Besides, since the hidden states h of the NNLM contains historical information, $W_{<i}$ can be substituted by h_i.

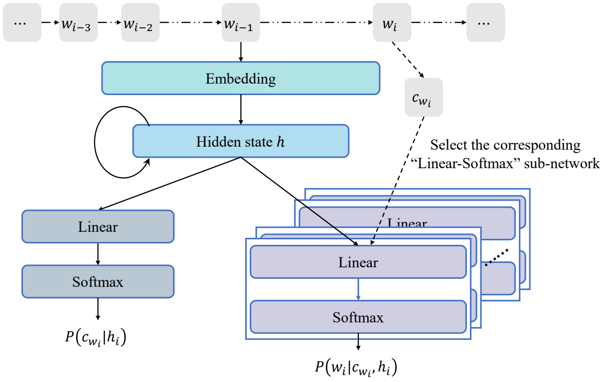


Figure 2: *The structure of class-based NNLM.*

Two conditional probabilities in the right-hand side of formula 2, the probability of current class c_{w_i} and the probability of present word w_i given c_{w_i} with historical information, can be respectively expressed by two sub-networks having the "Linear-Softmax" structure. The corresponding architecture is shown in Figure 2.

3. Clustering method

In [17], words that have similar frequency would be mapped into the same class. However, better performance can be obtained if words in the same class have identical properties. Hence, both knowledge-driven and data-driven methods are used to determine the classes of NNLM.

3.1. Knowledge-driven clustering method

Using extra semantic knowledge can improve performance in many tasks, including word clustering. Fortunately, the words can be clustered with some existing databases organizing words using semantic knowledge. The following narrative focuses on Chinese, and the approach is similar in other languages as long as having knowledge databases.

One of the most popular knowledge databases in Chinese is called "Synonym Cilin" [18] (starting now referred to as Cilin), which is an n-ary tree with a maximum depth of 5 (assuming the root node depth is 0). Each non-leaf node of Cilin represents a class containing similar words, and the children nodes make a more detailed division of the parent node. That is, as the depth increases, the words within the node become more similar. Since Cilin has already classified common words soundly, the vocabulary can be organized with the help of Cilin.

Because the vocabulary of Cilin is fixed, the morphological characteristics are used to cluster the words excluded from Cilin. Naturally, there are about 3,000 common characters, but there are tens of thousands of common words. In other words, most words are composed of more than one character. According to [14, 19], sub-words can represent the meaning of words to some extend. Therefore, we can decompose a word into a sub-word set and use it to find the word's semblable class in Cilin.

Before going into the specifics, some metrics related to Cilin will be defined. First, we use information content (IC) [20] to describe the amount of information contained in node u:

$$IC(u) \;= 1 - \frac{\log\left(hypo(u) + 1\right)}{\log\left(hypo(root) + 1\right)}, \tag{3}$$

where $hypo(u)$ denotes the number of hyponymy nodes below the node u, and $hypo(root) + 1$ is equivalent to the number of all nodes in Cilin. It can be seen that the deeper the nodes, the larger the value of the IC. From another perspective, the IC reflects the specificity of the node.

Consider two nodes in Cilin, u and v, and their paths to the root node must intersect at a certain node a, which is called the last common ancestor (LCA) of u and v, denoted as $LCA(u, v)$. Due to the nature of Cilin, the intersection node contains the information of both nodes, so $IC(a)$ can be regarded as the commonality between them. Therefore, we calculate the distance between two nodes by:

$$dis(u, v) = IC(u) + IC(v) - 2IC(LCA(u, v)). \tag{4}$$

The "similarity" of nodes, the opposite concept of "distance", is defined as follow:

$$sim(u, v) = \frac{D - dis(u, v)}{D}, \tag{5}$$

where D represents the maximum distance of nodes in Cilin.

With the above definitions, we can cluster the vocabulary. First, a proper depth of Cilin d is specified, and each node with depth d represents one class. Then, the vocabulary is divided into two categories based on whether the word appears in Cilin. For words that appear in Cilin, we can directly determine their classes. While for the word absent in Cilin, the sub-word set is used to estimate its class. For example, the word w, not belonging to Cilin, has the sub-word set $S = \{w_{i:j} | 0 \le i < j \le length(w)\}$. The closest node to w can be found using formula 6, where w' is the sub-word of w and $node(w')$ represents the Cilin node containing w'. Suppose the initial node is the root node, and by applying the above formula d times, the corresponding class of the final node is what the word w will belong to.

$$v = \arg\max_{v' \in V} \frac{1}{|S|} \sum_{w' \in S} sim(node(w'), v') \tag{6}$$

3.2. Data-driven clustering method

The advantage of the knowledge-driven clustering method is that it only needs a knowledge database and a small number of calculations to find a class. Meanwhile, this method solely focuses on the morphological characteristics of words not included in Cilin, so the method is not affected by context. However, this method needs an extra knowledge database, which may not be suitable for all text corpus.

Unlike the knowledge-driven clustering method, the data-driven clustering method learns the semantic representation of words from a large amount of text. A common approach is to embed semantic information in a low-dimensional dense space so that each word has a low-dimensional vector to represent its semantics. The distance between the vectors can reflect the semantic similarity between the words. Here we choose the word2vec [21] as the algorithm for extracting semantics.

After obtaining the word embedding vectors, the words can be clustered according to these word embedding vectors. We first determine the number of classes and then use the k-means clustering algorithm to cluster the vectors. Because the word2vec has embedded similar semantic words into the adjacent space, the k-means algorithm can assign similar words to the same class.

4. OOV word classification and parameter estimation

This part will introduce using the class-based NNLM to estimate OOV word parameters and calculating reasonable probability in the rescoring stage. There are three significant steps in our strategy:

1. Estimate the class $c_{w_{OOV}}$ of the OOV word w_{OOV} in the class-based NNLM
2. Estimate the parameters of w_{OOV} using the parameters of words in $c_{w_{OOV}}$
3. Calculate the probability of w_{OOV} using new estimated parameters

4.1. OOV word classification method

In section 3, we have introduced two ways to classify words. The knowledge-driven classification method has discussed how to deal with OOV words. The method, which only focuses on OOV words' morphological characteristics, can be easily used in classifying OOV words in the rescoring stage.

However, the data-driven classification method requires additional corpus to determine word information. Considering that the n-best sentences contain many errors, it is difficult to determine the OOV word's class using these uncertain sentences. Therefore, we decide to use the trained class-based NNLM itself to choose the OOV word's class. Note that the calculation of $P(c_{w_i}|h_i)$ in formula 2 has nothing to do with the current word. Besides, the trained NNLM can predict the class to which w_i belongs. Therefore, we can select the class with the highest confidence as w_i's class. That is:

$$c_{w_i} = \arg\max_c P(c|h_i). \tag{7}$$

4.2. OOV word parameter estimation

In class-based NNLM, each word's parameters include the embedding layer's parameters and the linear layer' parameters, and the latter is related to the category to which it belongs. Assuming that the class corresponding to the OOV word w_{OOV} is c, and there are k words in this class, we can weight the parameters of these k words to obtain w_{OOV}'s parameters. Let $\mathcal{W}_i$ denote the parameters of word w_i, then:

$$\mathcal{W}_{OOV} = \frac{1}{\sum_{i=1}^{k} \lambda_i} \sum_{i=1}^{k} \lambda_i \mathcal{W}_i, \tag{8}$$

where λ_i is the weight of w_i.

If λ for every word in c is equal to 1, the estimated parameters are the arithmetic mean of all words' parameters. Nevertheless, OOV words have low frequency, so we should adjust the weights so that the parameters of the OOV words predicted by the model are similar to the low-frequency words in the current class. We use the reciprocal of word frequency plus one (prevent division by 0 errors) as the weight. Let $count(w)$ be the number of times the word w appears in the training corpus, and we define the weight as:

$$\lambda_i = \frac{1}{count(w_i) + 1} \tag{9}$$

When the training process is over, all parameters have been fixed, and the frequency of each word is also constant, so the parameters obtained by weighted estimation are also determined. From the perspective of efficient calculation, we can pre-calculate each class's OOV word' parameters to avoid duplicate estimates in the rescoring stage and improve rescoring speed.

With the OOV parameters $\mathcal{W}_{OOV}$ obtained, three methods can be used to calculate the score during rescoring:

Prediction $\mathcal{W}_{OOV}$ is temporarily obtained to calculate the probability of the current OOV word, and then $\mathcal{W}_{OOV}$ will be dropped to keep the NNLM unchanged.

Extension Extend the vocabulary of NNLM when occurring OOV words and uses the estimated parameters to fill the parameters of the embedding layer and the final linear layer.

Replacement Substitute the w_{OOV} with some word w in the $c_{w_{OOV}}$. This method can avoid modifying the model, and the replacement can be applied on the fly. In general, the low-frequency words in $c_{w_{OOV}}$ are better choices.

5. Experiments

5.1. Experiments setup

The ASR system used in the experiment is trained on the AIShell dataset [22], which contains 178 hours of audio data from 11 fields. The rescoring class-based NNLM is trained on another dataset selected from CLMAD [23]. We choose 2,127,942 sentences that do not contain words outside the AIShell from the CLMAD's training set to form our training set S_T. S_T includes 41,215 different words. We also select 8,000 rows to create our validation set S_V.

The experiment involves four vocabularies, the size of which are 20k, 30k, 40k, and 44k, respectively. The first three vocabularies are obtained by sorting the word frequency from high to low and then choosing the words with the highest frequency. The last vocabulary, called the oracle vocabulary, is based on the vocabulary of size 40k and adds other words that appear in the AIShell. At the same time, another training data S_O containing S_T and all the sentences in the AIShell is also generated. Since the language model trained on S_O with the oracle vocabulary have no vocabulary mismatch problem, we call this model the oracle NNLM and regard the result of this language model as the upper bound of our experiment. Details of the data and the OOV rate under different vocabularies are shown in table 1.

Table 1: *OOV rate of different datasets.*

Dataset	**#tokens**	**Vocab. Size**		
		20k	30k	40k
S_T	21.7M	0.520%	0.086%	0.006%
S_V	77.6K	0.501%	0.053%	0.003%
S_O	22.8M	0.721%	0.149%	0.011%

All NNLMs have a single-layer LSTM, where the word vector dimension and the hidden layer dimension are both 128. The knowledge-driven clustering method specifies the Cilin depth as 2, corresponding to 96 classes. The algorithm of word2vec used in the data-driven clustering method is the continuous bag-of-word model. The vector dimension of word2vec is 128, and the window size is 5. Meanwhile, to make the results comparable, the number of clusters in the k-mean algorithm is the same as the knowledge-driven method.

The metric of the experiments is CER. Different models are firstly evaluated on the 50-best sentences generated by the ASR system. Then the best sentences, which own the highest score interpolated by the language model score and acoustics score, will be selected to calculate the CER on the golden sentence. It is worth mentioning that the CER of the char-based NNLM having 4k common characters is **13.26%** and the oracle NNLM's performance is significantly better, whose CER comes **10.57%**. Besides, since few works focus on addressing OOV words in the rescoring stage, we use the most common way, which is the normal NNLM with UNK token as our baseline. The baseline NNLM haves the same configuration except replacing the class-based softmax output layer with the standard softmax output layer.

5.2. Results and analyze

Table 2: *CER (%) results. The rows leading by "Know." show results using the knowledge-driven method. The rows leading by "Data" show results using the data-driven method. The CER of char-based NNLM with 4k common characters is 13.26%*

Model		**Vocab. Size**		
		20K	30K	40K
Baseline NNLM		14.03	12.60	11.33
Know.	Class-based NNLM	14.01	12.68	11.38
	Cilin-prediction	13.15	12.49	11.47
	Cilin-extension	12.92	12.43	11.57
	Cilin-replacement	**11.20**	**10.86**	**10.77**
Data	Class-based NNLM	13.94	12.50	**11.26**
	W2V-prediction	13.10	12.36	11.27
	W2V-extension	**12.66**	**12.15**	11.33
	W2V-replacement	12.96	12.32	11.42

Table 2 illustrates the CER results of using the knowledge-driven and data-driven OOV estimating method. It can be seen that using a small vocabulary will meet a severe vocabulary mismatch problem, which will lead to worse CER than the result of the char-based NNLM. Besides, without any optimization, the performance of the class-based NNLM is equivalent to that of normal NNLM.

However, benefiting from the methods proposed in this paper, the rescoring results of class-based NNLM have significantly been improved. For the knowledge-driven method, the replacement method gives the best results. The results of 30k and 40k vocabulary are even closed to the oracle model results. The improvement of methods "Cilin-prediction" and "Cilin-extension" relative to the baseline is not apparent. We believe that some classes contain many high-frequency words, causing the weighted parameters close to high-frequency words'.

For the data-driven approach, the replacement approach only reached a level comparable to the baseline due to OOV word clustering's poor accuracy compared to the knowledge-driven approach. However, the prediction and extension methods give better results than knowledge-driven methods, especially for small vocabulary experiments. Therefore, when a sound knowledge base is lacking, the data-driven method is still the right choice.

Table 3: *CER (%) results on larger vocabulary NNLM.*

Model	**Vocab. Size**			
	40k	60k	80k	100k
NNLM	15.13	14.89	14.69	14.54
NNLM (Cilin)	**12.42**	**12.02**	**11.86**	**11.80**

Further, we would like to explore the performance of rescoring language models possessing a larger vocabulary. This time, the training data is randomly selected from 14 fields of CLMAD in proportion, including 2 million rows and about 21 million words. It should be noted that in order to show how our method works in a more general context, the training data contains the words not included by the first-pass language model, that is, the relationship of vocabularies between the first-pass language model and the rescoring language models is the "intersection" relationship in Figure 1. As can be seen from the results in Table 3, expanding the vocabulary is helpful to reduce CER to some extent, but the effect still cannot make up for the performance loss caused by the vocabulary mismatch problem. However, the "Cilin-replacement" method, the strategy with the best overall performance in Table 2, can significantly close the CER gap with the oracle.

6. Conclusion

This paper introduces a paradigm to address the vocabulary mismatch OOV problem well. Firstly, the knowledge-driven method and data-driven method that can divide the vocabulary into meaningful classes are employed. Then the class-based NNLM is trained with classified vocabulary. During the rescoring stage, a proper class is assigned for each OOV word and estimates its parameters using the words in the corresponding class, giving the OOV word a more reasonable probability. The experiment shows that even if there is a large gap between rescoring NNLM vocabulary and ASR system vocabulary, our method can significantly reduce the CER.

7. Acknowledgements

This study was supported by Shanghai Municipal Science and Technology Major Project No.2021SHZDZX0102 and Xinhua News Agency No.SKLMCPTS2020003.

8. References

[1] R. J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent," in *Backpropagation: Theory, architectures, and applications*. Lawrence Erlbaum Associates, 1995, pp. 433–486.

[2] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] W. Naptali, "Study on n-gram language models for topic and out-of-vocabulary words," Ph.D. dissertation, Toyohashi University of Technology, 2011.

[5] F. Gallwitz, E. Noth, and H. Niemann, "A category based approach for recognition of out-of-vocabulary words," in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, vol. 1, 1996, pp. 228–231.

[6] T. Mikolov, I. Sutskever, A. Deoras, H. S. Le, S. Kombrink, and J. Cernocký, *Subword Language Modeling with Neural Networks*, 2012.

[7] C. Parada, M. Dredze, A. Sethy, and A. Rastrow, "Learning sub-word units for open vocabulary speech recognition," in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011, pp. 712–721.

[8] M. A. B. Shaik, A. E. D. Mousa, R. Schlüter, and H. Ney, "Hybrid language models using mixed types of sub-lexical units for open vocabulary german lvcsr," in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pp. 1441–1444.

[9] W. Naptali, M. Tsuchiya, and S. Nakagawa, "Class-based n-gram language model for new words using out-of-vocabulary to in-vocabulary similarity," *IEICE Transactions on Information and Systems*, vol. 95, no. 9, pp. 2308–2317, 2012.

[10] T. Müller and H. Schütze, "Improved modeling of out-of-vocabulary words using morphological classes," in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011, pp. 524–528.

[11] M. Singh, C. Greenberg, Y. Oualil, and D. Klakow, "Sub-word similarity based search for embeddings: Inducing rare-word embeddings for word similarity tasks and language modelling," in *Proceedings International Conference on Computational Linguistics (COLING)*, 2016, pp. 2061–2070.

[12] A. C. Limeres, F. F. Martínez, R. S. Segundo, and J. F. López, "Attention-based word vector prediction with lstms and its application to the oov problem in asr," in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 3520–3524.

[13] Y. Khassanov, Z. P. Zeng, V. T. Pham, H. H. Xu, and E. S. .Chng, "Enriching rare word representations in neural language models by embedding matrix augmentation," in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 3505–3509.

[14] R. N. Chen and K. Yu, "Fast oov words incorporation using structured word embeddings for neural network language model," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6119–6123.

[15] X. H. Zhang, D. Povey, and S. Khudanpur, "Oov recovery with efficient 2nd pass decoding and open-vocabulary word-level rnnlm rescoring for hybrid asr," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6334–6338.

[16] S. L. Zhang, H. Jiang, M. B. Xu, J. F. Hou, and L. R. Dai, "The fixed-size ordinally-forgetting encoding method for neural network language models," in *Proceedings of Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2015, pp. 495–500.

[17] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, "Extensions of recurrent neural network language model," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5528–5531.

[18] Q. Peng, X. H. Zhu, Y. S. Chen, L. Sun, and F. Li, "Ic-based approach for calculating word semantic similarity in cilin," *Application Research of Computers*, vol. 35, no. 2, pp. 400–404, 2018.

[19] Y. Li, B. Yu, M. Xue, and T. Liu, "Enhancing pre-trained chinese character representation with word-aligned attention," 2020.

[20] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in wordnet," in *Proceedings of European Conference on Artificial Intelligence (ECAI)*, 2004, pp. 1089–1090.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv:1301.3781*, 2013.

[22] H. Bu, J. Y. Du, X. Y. Na, B. G. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Proceedings of Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, 2017, pp. 58–62.

[23] Y. Bai, J. H. Tao, J. Y. Yi, Z. Q. Wen, and C. H. Fan, "Clmad: A chinese language model adaptation dataset," in *Proceedings of International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2018, pp. 275–279.