



Enriching Rare Word Representations in Neural Language Models by Embedding Matrix Augmentation

Yerbolat Khassanov¹, Zhiping Zeng², Van Tung Pham^{1,2}, Haihua Xu², Eng Siong Chng^{1,2}

¹School of Computer Science and Engineering, Nanyang Technological University, Singapore

²Temasek Laboratories, Nanyang Technological University, Singapore

{yerbolat002, zengzp, vantung001, haihuaxu, aseschnj}@ntu.edu.sg

Abstract

The neural language models (NLM) achieve strong generalization capability by learning the dense representation of words and using them to estimate probability distribution function. However, learning the representation of rare words is a challenging problem causing the NLM to produce unreliable probability estimates. To address this problem, we propose a method to enrich representations of rare words in pre-trained NLM and consequently improve its probability estimation performance. The proposed method augments the word embedding matrices of pre-trained NLM while keeping other parameters unchanged. Specifically, our method updates the embedding vectors of rare words using embedding vectors of other semantically and syntactically similar words. To evaluate the proposed method, we enrich the rare street names in the pre-trained NLM and use it to rescore 100-best hypotheses output from the Singapore English speech recognition system. The enriched NLM reduces the word error rate by 6% relative and improves the recognition accuracy of the rare words by 16% absolute as compared to the baseline NLM.

Index Terms: rare words, word embeddings, neural language models, speech recognition

1. Introduction

The neural language models (NLM) have achieved great success in many speech and language processing applications [1, 2, 3]. Particularly, it is highly employed in automatic speech recognition (ASR) systems to rescore the n -best hypotheses list where the state-of-the-art results are attained. Different from the traditional count-based N -gram models that suffer from the data sparsity problem [4], the NLMs possess superior generalization capability. The generalization is mainly achieved by learning dense vector representations of words as part of the training process and using them to express the probability function [1]. As a result, the learned word representations capture differences and commonalities between words, and thus enable NLMs to model different combination of words including the ones unseen during the training.

However, this concept assumes that each word appears a sufficient amount of times in the training data. For rare words with little or no training samples, the learned representations will be poor [5]. Consequently, the NLM will assign them unreliable probability estimates. Moreover, the representation of the rare word will be used as a context for the neighbouring words, as such, the entire word sequence containing the rare word will be underestimated. The problem exacerbates when a rare word is a named entity such as names of persons, locations, organizations and so on which are important keywords for downstream tasks such as voice search [6].

Currently, a common practice in language modeling is to ignore the rare word problem. For example, by limiting the NLM's vocabulary set to the most frequent words and treat the remaining words as out-of-vocabulary (OOV), i.e. mapping them to special $\langle unk \rangle$ token [7] or train the NLM with full vocabulary as usual. The former approach conflates all the meanings of rare words into a single representation, losing the properties of individual words. The latter approach will result in low-quality rare word representations. For both approaches, the probability estimates of hypotheses incorporating the rare words will be unreliable, leading to the sub-optimal performance of NLM.

In this work, we propose an efficient method to enrich the vector representations of rare words in pre-trained NLMs. The proposed method augments the word embedding matrices of pre-trained NLM while keeping other parameters unchanged. Specifically, our method shifts the rare word representation towards its semantic landmark in the embedding space using representations of other semantically and syntactically similar words. This method has been shown effective for word similarity task [8] and vocabulary expansion for NLMs [9]. We further extend its application to the rare word representation enrichment. To evaluate the proposed method, we first enrich the representations of rare Singapore street names in pre-trained NLM and then use it to rescore the 100-best hypotheses output from the state-of-the-art Singapore English ASR system. The enriched NLM reduces the word error rate by 6% relative and improves the recognition accuracy of the rare words by 16% absolute as compared to the strong baseline NLM.

The rest of the paper is organized as follows. Section 2 reviews related approaches designed to deal with the rare word problem. In Section 3, we briefly describe the architecture of baseline NLM. Section 4 presents the proposed embedding matrix augmentation technique. In Section 5, we explain the experiment setup and discuss the obtained results. Lastly, Section 6 concludes the paper.

2. Related works

The continuous vector representations of words are typically derived from large unlabeled corpora using co-occurrence statistics [10, 11]. They became the dominant feature for many natural language processing applications achieving the state-of-the-art results. To generalize well, however, these tasks require many occurrences of each word and fall short if a word appears only a handful of times [12]. Several approaches have been proposed to deal with the rare word problem and most of them can be classified under one of the three main categories shown below.

1) Morphological word representations. A bunch of proposed works resorts to subword level linguistic units by break-

ing down the words into morphemes [12, 13, 14]. For example, [12] represented words as a function of its morphemes where the recursive neural network is applied over morpheme embeddings to obtain the embedding for a whole word. While such works have been proven effective to deal with the infrequent word variations, they depend on the morphological analyzer such as *Morfessor* [15] and unable to model words whose morphemes are unseen during the training stage.

2) Character-level representations. To alleviate the rare word problem, finer level linguistic units such as syllables and characters have been also studied [16, 17, 18, 19]. For example, [16] explored both word-syllable and word-character level hybrid NLMs where most frequent words are kept unchanged, while rare words are split into the syllables and characters, respectively. In similar fashion, [17] and [18] examined character-aware NLM architectures which rely only on character level inputs, but predictions are still made at the word level.

Character-level models eliminate the need for morphological tagging or manual feature engineering and they comprise substantially fewer number of parameters compared to word-level models. Moreover, these approaches succeed at capturing properties of morphologically related words (e.g. ‘run’ vs ‘running’), but may fail to capture distinctions between semantically unrelated words (e.g. ‘run’ vs ‘rung’) [19, 5].

3) Knowledge-powered word representations. Another direction of works leverage external knowledge to enhance representations of rare words [5, 20, 21]. For example, [5] employed word definitions obtained from the *WordNet* [22] to model rare words on the separate network. Alternatively, [20] proposed to incorporate external knowledge as a regularization term to the original model’s objective function. Although these approaches have shown promising results, they highly depend on the availability of external hand engineered lexical resources.

Note that the aforementioned approaches can be also used jointly. For example, by using factored NLM architecture [23] where different feature types can be combined.

3. Baseline NLM architecture

The NLM architectures can be generally classified into two main categories: feedforward [1] and recurrent [2]. Our method can be applied to both of them, but in this paper we will focus on recurrent architecture with LSTM units which has been shown to achieve the state-of-the-art results [24].

The conventional recurrent LSTM architecture can be decoupled into three main components as shown in Figure 1: 1) input projection layer, 2) middle layers, and 3) output projection layer. The input layer is parameterized by input embedding matrix \mathbf{S} used to map one-hot encoding representation of word $w_t \in \mathbb{R}^{|V|}$ at time t into continuous vector representation s_t , where $|V|$ is a vocabulary size:

$$s_t = \mathbf{S}w_t \quad (1)$$

The embedding vector s_t and a high-level context feature vector from the previous time step h_{t-1} are then combined by non-linear middle layers, which can be represented as function $f()$, to produce a new context feature vector h_t :

$$h_t = f(s_t, h_{t-1}) \quad (2)$$

The non-linear function $f()$ can employ simple activation units such as ReLU and hyperbolic tangent or more complex units such as LSTM and GRU. The middle layers can be also formed by composing several such functions.

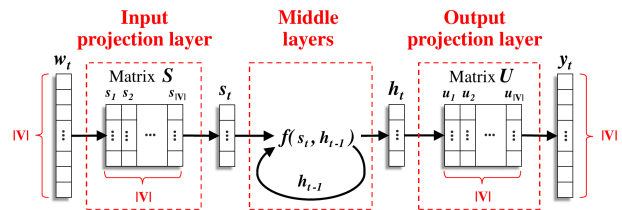


Figure 1: NLM architecture decomposed into three components.

Lastly, the context vector h_t is fed to the output layer which is parameterized by output embedding matrix \mathbf{U} to produce a high-dimensional vector $y_t \in \mathbb{R}^{|V|}$:

$$y_t = \mathbf{U}^T h_t \quad (3)$$

The entries of output vector y_t represent the scores of words to follow the context h_t . These scores are then normalized by softmax function to form a probability distribution.

Our method modifies the embedding matrices \mathbf{S} and \mathbf{U} while keeping the middle layer $f()$ intact as will be explained in the next section.

4. Embedding matrix augmentation

We start with the assumption that we are given a pre-trained NLM which models full vocabulary including both frequent and rare words. In such models, the rare words will be poorly represented, leading to the sub-optimal performance. Therefore, our goal is to enrich the representation of rare words without collecting additional training data and incurring expensive post-processing procedures.

To achieve this goal, we exploit the structure of NLM where input and output layers are parameterized by word embedding matrices (see Figure 1). Particularly, we propose to modify both input and output embedding vectors of the rare words, while keeping the parameters of middle layers unchanged. The embedding vectors of the rare words are modified using embedding vectors of other semantically and syntactically similar words. This approach will retain the linguistic regularities encapsulated within original pre-trained NLM, given that embeddings of the rare words are properly modified. Our method can be also viewed as a language model adaptation task [25] where instead of topic or speaking style the vocabulary is adapted to conform with the words used in the target domain.

The proposed method has three main steps: 1) identifying the rare words, 2) finding similar words and 3) enriching rare word representations.

1) Identifying the rare words. To identify the rare words we can simply count the frequency of words in the training data and set a frequency threshold below which all words are considered rare. This approach, however, might result in too many rare words. To reduce the computation time, we can limit the rare words to those which appear in the n -best hypotheses or word lattice output.

2) Finding similar words. Given a subset of rare words $V_{rare} \subset V$, the next step is to select a list of similar candidate words \mathcal{C} for each rare word. The selected candidates will be used to enrich representations of rare words, hence, they must be frequent and present in the vocabulary V of NLM. In addition, they should be similar to the target rare word both in semantic meaning and syntactic behavior. Note that selecting inadequate candidates might deteriorate the performance of NLM, thus, they should be carefully inspected.

Several effective methods exist that can find appropriate candidate words. For example, using lexical resources that contain synonyms and related words such as WordNet or employing pre-trained word embeddings from skip-gram or cbow models [10] which can also find similar words. In our experiments, we use lexical resource containing a list of Singapore street names where frequent street names will be used to update representations of rare street names.

3) Enriching rare word representations. Let s_r be an embedding vector of some rare word w_r in space defined by input embedding matrix \mathbf{S} and let \mathcal{C}_r be corresponding set of similar words. We enrich the representation of s_r using the words in \mathcal{C}_r by the following formula:

$$\hat{s}_r = \frac{s_r + \sum_{s_c \in \mathcal{C}_r} m_c s_c}{|\mathcal{C}_r| + 1} \quad (4)$$

where \hat{s}_r is the enriched representation of s_r , s_c is an embedding vector of similar candidate word and m_c is a metric used to weigh candidates based on importance. The m_c can be estimated using frequency counts or similarity score where most frequent or most similar candidates are given higher weights. In our experiments, we weigh the candidates equally.

The Eq. (4) typically shifts the embedding of a rare word towards the weighted centroid of its semantic landmark. The motivation is that highly correlated words, in terms of both semantic meaning and syntactic behavior, should be close to each other in the embedding space. We then use the same candidates and formula to update the corresponding rare word embedding u_r in the output embedding matrix \mathbf{U} . This procedure is then repeated for the remaining words in subset V_{rare} .

5. Experiment

In this section, we describe experiments conducted to evaluate the effectiveness of the proposed embedding matrix augmentation technique. Particularly, we first enrich the rare Singapore street name representations in pre-trained NLM and then use the enriched NLM to rescore 100-best hypotheses output from the Singapore English ASR. The ASR system is built by Kaldi [26] speech recognition toolkit using Singapore English speech corpus. To highlight the importance of enriching the rare word representations, we used 1 hour recording of 9 read articles about Singapore streets as an evaluation set¹ (7.3k words).

We compare our enriched model against three state-of-the-art language models (LM) including Kneser-Ney smoothed 4-gram² (KN4), Kaldi-RNNLM [27] and recurrent LSTM LM [3]. Our model is obtained by enriching the representations of rare Singapore street names in the recurrent LSTM LM, and we call it E-LSTM. The performance of these four LMs is evaluated on the 100-best rescoring task.

5.1. Experiment setup

Acoustic model. The acoustic model (AM) is built using ‘nnet3+chain’ setup of Kaldi and trained on 270 hours of transcribed Singapore English data which mostly consist of speech taken from parliament, talk shows and interviews.

Lexicon. The lexicon is constructed by assembling 51k unique words which include around 2k Singapore street names. To avoid ambiguity, the street names consisting of more than

one word are joined using the underscore symbol, e.g. ‘Boon lay’ is changed to ‘Boon_lay’. This lexicon was also used as a vocabulary set for LMs.

Language model. To train LMs, we used AM transcripts and web crawled Singapore related data which resulted in total 1M in-domain sentences (16M words). In addition, we used Google’s 1 billion word (1BW) benchmark corpus [28] to account for generic English word sequence statistics.

The KN4 is trained on combined in-domain (AM transcript+web crawled) and generic 1BW data. It was built using SRILM toolkit [29] with 51k vocabulary set. We used KN4 model to rescore both word lattice and 100-best list. Its pruned version KN4_pruned was used during the decoding stage.

The Kaldi-RNNLM is a word-character level hybrid model designed to overcome the rare word problem by decomposing the rare words into character n -grams while keeping the most frequent words unchanged. It was trained as a 2-layer LSTM³ with 800 units in each layer using only in-domain data. The input and output embedding matrices were tied and embedding space dimension was set to 800. For vocabulary, we tried to keep a different number of most frequent words and found 20k to perform best, the remaining 31k words were decomposed.

The recurrent LSTM LM is a word level model which was built using our own implementation in PyTorch [30]. It was trained as a single layer LSTM with 1k units using in-domain data and 51k vocabulary set. The input and output embedding space dimensions were set to 300 and 1000, respectively. The parameters of the model were learned by truncated BPTT [31] and SGD with gradient clipping. We also applied dropout for regularization [32].

Lastly, our E-LSTM model is obtained by enriching the rare word representations in the pre-trained recurrent LSTM LM. As a case study, we use Singapore street names⁴ where frequent streets will be used to enrich the rare street representations. In particular, we first count the frequency of each street name in the in-domain data and then divide them into two subsets of frequent and rare streets using some threshold value. Next, we randomly choose words from the subset of frequent streets and employ them to enrich the representations of all rare streets (allStreets) using the Eq. (4). To reduce computation time, we also tried to enrich only rare streets present in the 100-best hypotheses output (fromNbest).

5.2. Experiment results

The experiment results are shown in Table 1. In these experiments, we divide the street names into frequent and rare subsets using the threshold value of 10. To enrich the rare streets we used 5 randomly chosen frequent street names⁵. The initial word error rate (WER) without any rescoring is 17.07%.

The obtained results show that the E-LSTM model outperforms the strong KN4 used to rescore the word lattice by 16% relative WER (from 16.52% to 13.83%). Moreover, it achieves 6% relative WER improvement over Kaldi-RNNLM and LSTM models (from 14.73% to 13.83%). We found that enriching only rare streets present in the 100-best hypotheses (*fromNbest*) achieves a similar result as enriching all rare streets (*allStreets*), while *fromNbest* being much faster.

The state-of-the-art WER results are usually achieved by interpolating NLM and count-based N -gram models which have

¹https://github.com/khassanoff/SG_streets

²We also examined other N -gram models and found 4-gram to work best for our case.

³Changing the number of layers and its size didn’t improve WER.

⁴<https://geographic.org/streetview/singapore/>

⁵For consistency, we fix the chosen frequent streets to be same.

Table 1: The perplexity and WER results on evaluation set

LM	Perplexity	Rescore	WER
KN4_pruned	436	-	17.07%
KN4	351	Lattice	16.52%
		100-best	16.84%
Kaldi-RNNLM	-	100-best	14.73%
+KN4	-	100-best	14.10%
LSTM	295	100-best	14.74%
+KN4	-	100-best	14.95%
E-LSTM (allStreets)	242	100-best	13.87%
+KN4	-	100-best	13.58%
E-LSTM (fromNbest)	234	100-best	13.83%
+KN4	-	100-best	13.55%

been shown to complement each other [2, 3]. To this end, we interpolated⁶ NLMs with KN4 and achieved further WER reductions. Interestingly, the baseline LSTM model doesn't benefit from KN4⁷, while E-LSTM gains additional 2% relative WER improvement (from 13.83% to 13.55%).

5.2.1. Changing the frequency threshold value

To determine the effective frequency threshold range, used to split the street names into frequent and rare subsets, we repeat the experiment with different threshold values as shown in Figure 2. We observe that setting it between 5 and 50 is sufficient to achieve good results. On the other hand, setting it too low or high will deteriorate WER as can be seen from the left and right tails of the plot in Figure 2.

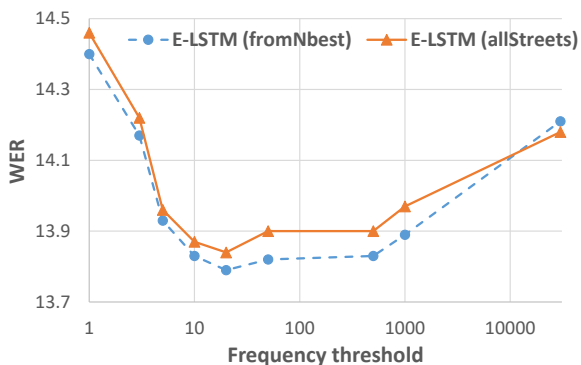


Figure 2: WER performance of E-LSTM at different frequency threshold values used to split frequent and rare street names.

5.2.2. Changing the number of frequent words

We also repeat the experiment to determine the optimal number of frequent words to use to enrich the rare words. We observed that for all cases the WER results are similarly good. For fast computation, we recommend to use around 3-20 most frequent words. This experiment is incomplete as we didn't examine the quality of selected frequent words which requires more substantial analysis. Due to the space limitations, we leave the further analysis for future work.

⁶Interpolation weight for KN4 is set to 0.3.

⁷Changing the interpolation weight didn't help.

Table 2: Recognition accuracy of 265 rare street names

LM	Rescore	Accuracy
KN4_pruned	-	37.36%
KN4	Lattice	36.98%
Kaldi-RNNLM	100-best	43.40%
LSTM	100-best	43.02%
E-LSTM (fromNbest)	100-best	59.25%

5.2.3. Recognition accuracy of enriched rare words

To ensure that WER improvements are achieved as a result of correctly recognizing the enriched rare street names, we compute the recognition accuracy of 265 rare street names (see Table 2). The experiment results show that after enriching the baseline recurrent LSTM LM, the recognition accuracy is increased by 16.23% (from 43.02% to 59.25%) achieving the best result among all LMs. Furthermore, we observe that correctly recognizing the rare street names also helps to recover neighbouring words (see Table 3). These results confirm the effectiveness of the proposed method.

Table 3: Examples of correctly recovered neighbouring words after rescoring with E-LSTM

LM	Example
KN4_pruned	1) a hawker centre and market began operations at bully plays in nineteen seventy six 2) by nineteen ninety four when the book development guide plan was announced
LSTM	1) a hawker centre and market began operation at bully police in nineteen seventy six 2) by nineteen ninety four when the product development gap plan was announced
E-LSTM	1) a hawker centre and market began operations at boon lay place in nineteen seventy six 2) by nineteen ninety four when the bedok development guide plan was announced

6. Conclusions

In this work, we proposed an effective method to enrich the representations of rare words in pre-trained NLM. The proposed method augments the embedding matrices of pre-trained NLM while keeping other parameters unchanged. Importantly, it doesn't require additional in-domain data and expensive post-training procedures. We applied our method to enrich the rare Singapore street names in pre-trained LSTM LM and used it to rescore the 100-best list generated by the state-of-the-art Singapore English ASR system. The enriched LSTM LM achieved 6% relative WER improvement over the baseline LSTM LM. In comparison to other strong baseline LMs, our method achieves significant WER improvements, i.e. 6% and 16% improvement over Kaldi-RNNLM and KN4, respectively. Moreover, the enriched LSTM increased the recognition accuracy of rare street names by 16% absolute. We believe that the proposed method can benefit other models with similar network architecture and be easily adapted to other scenarios.

7. Acknowledgements

This work is supported by the project of Alibaba-NTU Singapore Joint Research Institute.

8. References

- [1] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.
- [2] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *11th Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2010.
- [3] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *13th Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2012.
- [4] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.
- [5] D. Bahdanau *et al.*, "Learning to compute word embeddings on the fly," *arXiv preprint arXiv:1706.00286*, 2017.
- [6] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, "Your word is my command: google search by voice: A case study," in *Advances in speech recognition*. Springer, 2010, pp. 61–90.
- [7] J. Park, X. Liu, M. J. Gales, and P. C. Woodland, "Improved neural network based language modelling and adaptation," in *11th Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2010.
- [8] M. T. Pilehvar and N. Collier, "Inducing embeddings for rare and unseen words by leveraging lexical resources," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL*, 2017, pp. 388–393.
- [9] Y. Khassanov and C. E. Siong, "Unsupervised and efficient vocabulary expansion for recurrent neural network language models in ast," in *19th Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018, pp. 3343–3347.
- [10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [11] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, 2014, pp. 1532–1543.
- [12] T. Luong, R. Socher, and C. Manning, "Better word representations with recursive neural networks for morphology," in *Proceedings of the 17th Conference on Computational Natural Language Learning, CoNLL*, 2013, pp. 104–113.
- [13] A. Lazaridou *et al.*, "Compositionally derived representations of morphologically complex words in distributional semantics," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL, Volume 1: Long Papers*, 2013, pp. 1517–1526.
- [14] S. Qiu, Q. Cui, J. Bian, B. Gao, and T.-Y. Liu, "Co-learning of word representations and morpheme representations," in *25th International Conference on Computational Linguistics, COLING*, 2014, pp. 141–150.
- [15] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *TSLP*, vol. 4, no. 1, p. 3, 2007.
- [16] T. Mikolov, I. Sutskever, A. Deoras, H.-S. Le, and S. Kombrink, "Subword language modeling with neural networks," 2012.
- [17] W. Ling *et al.*, "Finding function in form: Compositional character models for open vocabulary word representation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, 2015, pp. 1520–1530.
- [18] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proceedings of the 13th AAAI Conference on Artificial Intelligence*, 2016, pp. 2741–2749.
- [19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *TACL*, vol. 5, pp. 135–146, 2017.
- [20] C. Xu *et al.*, "Rc-net: A general framework for incorporating knowledge into word representations," in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, CIKM*, 2014, pp. 1219–1228.
- [21] M. Faruqui *et al.*, "Retrofitting word vectors to semantic lexicons," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT*, 2015, pp. 1606–1615.
- [22] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [23] A. Alexandrescu and K. Kirchhoff, "Factored neural language models," in *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. ACL, 2006.
- [24] M. Sundermeyer *et al.*, "Comparison of feedforward and recurrent neural network language models," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2013, pp. 8430–8434.
- [25] Y. Khassanov *et al.*, "Unsupervised language model adaptation by data selection for speech recognition," in *Asian Conference on Intelligent Information and Database Systems, ACIIDS*. Springer, 2017, pp. 508–517.
- [26] D. Povey *et al.*, "The kaldi speech recognition toolkit," in *IEEE Signal Processing Society*, 2011.
- [27] H. Xu *et al.*, "Neural network language modeling with letter-based features and importance sampling," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2018, pp. 6109–6113.
- [28] C. Chelba *et al.*, "One billion word benchmark for measuring progress in statistical language modeling," in *15th Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2014, pp. 2635–2639.
- [29] A. Stolcke, "Srlm—an extensible language modeling toolkit," in *7th International Conference on Spoken Language Processing, ICSLP-INTERSPEECH*, 2002.
- [30] A. Paszke *et al.*, "Automatic differentiation in pytorch," 2017.
- [31] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
- [32] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.