ENHANCE RNNLMS WITH HIERARCHICAL MULTI-TASK LEARNING FOR ASR

Minguang Song, Yunxin Zhao

Department of EECS, University of Missouri, Columbia, MO, USA

ABSTRACT

It is known that neural language models (NLMs) can implicitly learn certain linguistic information from text. While generally NLMs only use word feature input, the success of factored NLMs has indicated a benefit of using additional linguistic feature inputs for language modeling. On the other hand, multi-task learning (MTL) has shown positive effects on the generalization performance of various natural language processing (NLP) tasks, including language modeling. However, how to best share information among related tasks in MTL remains to be addressed. In this current work, we propose a hierarchical multi-task learning (HMTL) approach to incorporate linguistic knowledge into recurrent neural network language models (RNNLM), instead of using linguistic features as word factors. Specifically, we consider the auxiliary tasks of chunking, part of speech tagging, and named entity recognition, and supervise the learning of these auxiliary tasks in a hierarchical way. Our proposed method has the potential of helping language models learn knowledge of linguistic hierarchy from the auxiliary tasks, and improve the performance of RNNLMs on automatic speech recognition (ASR). We have evaluated our proposed HMTL method on WSJ and AMI speech recognition tasks. Our experiment results demonstrate the effectiveness of the proposed approach.

***Index Terms*—** recurrent neural network, language model, multi-task learning, linguistic hierarchy

1. INTRODUCTION

A language model (LM) predicts the probabilities of word sequences. In the deep learning era, Recurrent neural network language models (RNNLMs) [1] have been demonstrated to outperform traditional statistical n-gram language models. RNNLMs have been widely applied in the natural language processing (NLP) field [2, 3]. RNNLMs with gating mechanisms such as long short-term memory (LSTM) [4] can encode longer range dependencies than simple recurrent unit, and they have led to significant performance gains in language modeling for automatic speech recognition [5].

Typically, RNNLMs only use words as input feature. Although LSTM LMs could implicitly learn certain linguistic information such as syntax structure [6] from word feature inputs, there are more linguistic resources that can be helpful for language modeling. In factored NNLMs [7, 8, 9], various linguistic features have been explored, where topic features generated from latent Dirichlet allocation (LDA) [10] helped improve perplexities (PPLs) and word error rates (WERs) [8], and morphological, syntactic and semantic features serving as input factors of RNNLMs yielded significant WER reductions on speech recognition tasks [9]. In machine translation [2], part-of-speech (POS) tags and word clusters were employed, leading to improved BLEU scores.

Multi-task learning (MTL) [11] is a popular approach to training deep neural networks, and it offers a fruitful venue to exploit linguistic knowledge in NLP. The purpose of MTL is often to improve a system's generalization performance on a primary task by exploiting relevant information contained in some related auxiliary tasks through a joint training on these tasks. This approach has been extensively investigated and successfully applied to several NLP problems [2, 12, 13, 14]. One important question in MTL is how to share information among multiple tasks. On one hand, MTL is typically learned by a shared data representation, either hard or soft, with task-specific output layers located at the same level or in the outmost layer of a deep neural network. On the other hand, the work of [13] suggests that having all tasks supervised at the outmost level is often suboptimal and certain linguistic hierarchy should be considered when using MTL. Hierarchical MTL (HMTL) has been explored and proven to be effective in several NLP tasks [13, 14, 15]. Yet, the effectiveness of HMTL for the language modeling task has not been investigated.

In this current work, we propose a HMTL deep architecture to train RNNLMs. We propose MTL for RNNLM by using chunking, POS tagging, and named entity recognition (NER) as the auxiliary tasks, exploiting such linguistic hierarchy information to improve word prediction. Instead of learning the primary and the auxiliary tasks all at the same outmost layer of the neural network, we supervise the three auxiliary tasks at different levels of the network. Specifically, in our multi-layer LSTM network, a bottom network layer is used for supervising the phrase level task of chunking, the next layer for the word level task of POS, and the further next layer for NER that refines a category of POS. We have evaluated our HMTL approach on two tasks of automatic speech recognition: Wall Street Journal (WSJ) [16] task and the AMI [17] IHM task, with improved performances of WERs achieved on both tasks demonstrating the effectiveness of our approach.

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

The rest of the paper is organized as follows. In Section 2 we briefly review multi-task learning. Our proposed HMTL method is described in Section 3. Experiment setup and results are shown and analyzed in Section 4. A conclusion is made in Section 5.

2. RELATED WORKS

MTL aims to improve a primary task's generalization performance by leveraging domain-specific information in related tasks. Deep neural networks trained with MTL methods have been successful in many NLP tasks, including POS tagging and chunking [12, 13, 15], named entity recognition (NER) [12, 14], machine translation [2], etc.

It is typical in MTL that a neural network is trained jointly with multi-task supervisions to learn a shared (hard or soft) representation. The primary and auxiliary tasks are supervised at the same level of the neural network – the outmost layer [7, 12, 18]. Nevertheless, for natural language processing, such a parallel structure ignores the information in linguistic hierarchies. Recently, it has been shown that supervising different auxiliary tasks in hierarchy leads to better performance. In [13], the authors improved the primary tasks of Chunking and CCG supertagging by supervising the auxiliary task of POS tagging at a low level. In [15], the authors improved five tasks of POS tagging, chunking, dependency parsing, semantic relatedness, and textual entailment by using a supervision order from the word level, to the syntactic level, and finally to the semantic level. In [14], four semantic tasks of NER, entity mention detection, coreference resolution, and relation extraction that share inter-dependencies were hierarchically supervised.

While MTL prospers in these NLP tasks, language modeling with MTL has not been well investigated. In [18], the auxiliary task of POS tagging was used in training a RNNLM to reduce word error rates on speech recognition tasks. In [2], the auxiliary tasks of POS tagging and word clustering were used in training factored RNNLMs to improve BLEU scores in machine translation. Neither of the two RNNLM works considered HMTL in their investigations.

3. PROPOSED METHOD

3.1. Model Architecture

Inspired by [14], we investigate using MTL in a hierarchical manner for RNNLM training. Here, the primary task is language modeling, which predicts a target word according to a word history. We use LSTM as the recurrent unit to encode history contexts in RNNLM. In order to learn better representations from linguistic knowledge and use it to regularize language model training, we adopt three auxiliary NLP tasks: chunking, POS tagging, and NER. In training the LSTM, we define the composite training loss function as:

$$L = L_{lm} + \sum_{i=1}^{T} \alpha_i L_{aux}(i) \qquad (1)$$

where L_{lm} is the loss for the language modeling task, $L_{aux}(i)$ is the loss for the i-th auxiliary task, α_i is the weight of $L_{aux}(i)$, and T is the total number of auxiliary tasks.

The architecture of our proposed model is shown in Fig. 1. The three NLP tasks are supervised in the order of chunking, POS tagging, and NER, from lower level to higher level of the model. The different linguistic knowledge is learned at the different stages of the information flow. Shortcut connections are considered to be crucial for hierarchical multi-task learning in NLP [14, 15]. Accordingly, we also adopt the shortcut connections from the output of the first LSTM layer to the higher LSTM layers so that the higher layers could directly access the lower layer representations. In our design, the representation vector from the short cut is fused with that produced by each downstream auxiliary LSTM in an additive manner. For example, the first LSTM layer output is added to the chunking LSTM layer output to serve as the input to the LSTM layer of POS, etc.

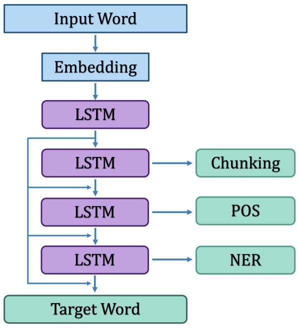

Fig. 1: RNNLM with hierarchical multi-task learning

3.2. Auxiliary Tasks

An example of our three auxiliary tasks is shown in Table 1. Chunking is a shallow parsing task which assigns a chunking tag to each word. The tag specifies for each word its position relative to the phrase span of interest in a sentence, together with the syntax tag of the phrase span. The tagging scheme used in our work is BIOES, which indicates whether a word begins a span (B), occurs inside a span (I), is out of any span of interest (O), ends a span (E), or is in a single-word span (S). For example, B-NP (as for the word ”New”) indicates that the current word is the beginning word of a noun phrase. There are a total of 42 BIOES labels in the chunking task [12]. The purpose of POS tagging is to classify each word with a tag of grammatical categories (e.g. noun, verb) in the 45-tag Penn Treebank tagset [19]. The task of NER is to label words of names in a sentence with 18 refined categories such as “PERSON” or “LOCATION”, where each word is assigned a tag that is prefixed by one of the five indicators as defined in the chunking task such as beginning or ending of a named entity.

word	John	lives	in	New	York
Chunking	S-NP	S-VP	S-PP	B-NP	E-NP
POS	NNP	VBZ	IN	NNP	NNP
NER	S-PERSON	O	O	B-GPE	E-GPE

Table 1: Example of the three auxiliary NLP tasks

Our objective here is to improve the word prediction performance of language model. Within the task spectrum of NLP, language modeling is a low-level task relative to other tasks involving high-level semantics. Therefore, the hierarchy of auxiliary tasks used in our RNNLM training is different from the previously reported NLP works of [13, 14, 15]. The three auxiliary NLP tasks introduced in training the RNNLM contain different aspects of linguistic knowledge. The chunking tags provide the phrasal context or scope for POS tags, the POS tags determine whether a word is likely a named entity or not, and the NER tags further state the named entity type of a name word.

Consider the word ”New” in Table 1. With its chunking tag ”B-NP” (beginning of noun phrase), POS tag ”NNP” (noun), and named entity label ”B-GPE” (beginning of geopolitical entity), the linguistic role of the word is refined in layers from coarse to detailed. By having such a rich representation for the word, the chance of predicting the next word being a noun of an entity name is likely high, and thereby increasing the probability of the next word for ”York.” In such a way, the word prediction task would benefit from the hierarchical linguistic properties of words learned in training the language model.

In summary, by injecting the explicit syntactic structure, syntactic role, and name type information hierarchically into network training, a language model will likely learn rich representations of words to encode linguistic properties that are relevant to word interactions. Finally, such linguistic knowledge enriched word representations will help a language model predict words more accurately.

We also tried other orders or hierarchies of these NLP tasks. Our experiment results confirmed that our proposed hierarchy of chunking → POS → NER performed the best among the other hierarchies. Due to space limitation, here we omit the detailed results of the other orders.

4. EXPERIMENTS

We evaluated our proposed HMTL method for language model training on two speech recognition tasks: the Wall Street Journal (WSJ) Corpus and the AMI meeting individual headset microphones (IHM) Corpus. The Kaldi toolkit [20] was used to train the acoustic models and to generate the word lattices as well as n-best (n=100) sentence hypotheses.

Since it was hard to obtain ground truth labels for the training data, without loss of generality, we first used Stanford CoreNLP [21] to restore the case information of texts, and then fed the cased texts into a state-of-the-art sequence labeling toolkit [22] to obtain the labels needed for the auxiliary NLP tasks. We set the weights α_i in Eq(1) of the auxiliary tasks to be equal for simplicity, and determined the value by a grid search in the range of [0.1, 1.0] with a step size of 0.1.

PyTorch [23] was used to train the RNNLMs. The size of the LSTM hidden layer was fixed to 200, and the steps of BPTT were set to 35 for WSJ and 15 for AMI.

In the following experiments, we report the results of WERs and perplexities (PPLs) of RNNLMs trained without MTL vs. with HMTL. We also compare the results with the case of having all auxiliary tasks supervised at the outmost layer, referred to as flat MTL (FMTL). In addition, ablation experiments were performed to assess the contributions by individual auxiliary tasks. In all of these cases, 4 LSTM layers were used in a RNNLM. For each case, we trained five models by using five random weight initialization, and we report the average results and the standard deviation from the five models.

4.1. WSJ

The standard WSJ training set with 40M words was used to train our RNNLMs. The RNNLMs were evaluated on the Eval 92 and Eval 93 test sets of the 20k-word WSJ task, with 333 sentences (5643 words) and 213 sentences (3448 words) in Eval 92 and Eval 93 sets, respectively. The baseline word error rate was generated by using the HMM-DNN acoustic model trained with the Kaldi recipe [24]. The DARPA trigram language model for WSJ was used for decoding and lattice generation, giving WERs of 5.78% for Eval 92 set and 7.34% for Eval 93 set. From the word lattices produced by the baseline system, we derived n-best sentence hypotheses with n = 100, and performed rescoring by using RNNLMs trained from the different strategies. The weights α_i's of the auxiliary tasks were set as 0.1. The RNNLMs were each interpolated with the DARPA trigram model.

4.1.1. Word error rates

	Eval 92	Eval 93
Baseline trigram	5.78	7.34
+RNNLM (w/o MTL)	4.38 ± 0.06	5.76 ± 0.15
+RNNLM (w/ FMTL)	4.27 ± 0.06	5.96 ± 0.14
+RNNLM (w/ HMTL)	**4.18 ± 0.08**	**5.57 ± 0.29**

Table 2: WER (%) results on the WSJ corpus.

The WER results are shown in Table 2. The RNNLMs without MTL largely reduced the WER over the baseline trigram language model. By using the RNNLMs with FMTL, the WERs were reduced on the Eval 92 set but increased on the Eval 93 set. On the other hand, the RNNLMs trained with HMTL reduced the WERs on both test sets. In comparison with RNNLMs without MTL, WER was reduced by 0.20% absolute and 4.57% relative on the Eval 92 test set, and by 0.19% absolute and 3.30% relative on the Eval 93 test set, and the WER reduction related to grammar/syntax was around 50%, and that related to names was around 16% (on Eval 92 and 93 sets combined). The WER results demonstrated that our proposed training method for RNNLMs learnt better representations by arranging the auxiliary task supervision in a hierarchy than flat at the same level.

It is worth noting that the impact of a LM on WER reductions is also limited by the oracle WER of the n-best hypothesis lists produced by speech decoding search. On the Eval 92 and 93 sets, the oracle WERs were 2.92% and 3.34%, respectively. As such, relative to RNNLM without MTL, the ranges of absolute WER reduction were only 1.46% and 2.42% on Eval 92 and 93 sets, where our HMTL LM actually reduced WER by 13.70% and 7.85% over the two ranges.

	Eval 92	Eval 93
chunking+POS+NER	4.18 ± 0.08	5.57 ± 0.29
chunking+POS	4.26 ± 0.05	5.81 ± 0.16
POS+NER	4.37 ± 0.08	5.78 ± 0.32
chunking+NER	4.26 ± 0.08	5.75 ± 0.32

Table 3: Ablation study of WER (%) results on WSJ corpus.

We performed an ablation study on the three auxiliary tasks by removing one of the three tasks at a time (without changing the number of LSTM layers). The results are shown in Table 3. It is observed that removing any one task would increase WERs on both test sets, where removing the chunking task had the largest negative impact on WERs, indicating this task contributed the most to the WER reductions in HMTL. The contributions of POS tagging and NER tasks were comparable and were both less than chunking.

4.1.2. *Perplexities*

	Eval 92	Eval 93
RNNLM (w/o MTL)	66.2 ± 0.4	66.7 ± 0.5
RNNLM (w/ FMTL)	65.8 ± 0.7	66.1 ± 0.7
RNNLM (w/ HMTL)	65.8 ± 0.7	66.0 ± 0.8

Table 4: Perplexity results on WSJ corpus

The perplexity results from the RNNLMs on the two test sets are shown in Table 4. We can see that on average, the PPLs were reduced only slightly by MTL. Generally, there was not much difference in PPL by the HMTL and FMTL methods. Since our focus is on speech recognition, the reduction on WER is the most important. Moreover, perplexity and WER are often not well correlated. This phenomenon has been observed by previous works in [18, 25, 26, 27, 28].

4.2. AMI

The AMI meeting corpus consisted of 100 hours of meeting recordings. The Kaldi AMI IHM chain model recipe [29] was used to train the acoustic model. The size of vocabulary for decoding was 49k. The training transcriptions, as well as the first part of the Fisher corpus, were used to train the language model. A pruned trigram language model was used for decoding and lattice generation. All the LMs were trained on the combined (Fisher+AMI) texts of 14M words. The weights of all auxiliary tasks were set as 0.3.

We evaluated the performance of RNNLMs on the development (Dev) set (95k words) and the evaluation (Eval) set (90k words). The vocabulary size of RNNLMs was 34k, which was generated by intersecting the decoding vocabulary with the words in the training data.

4.2.1. *Word error rates*

The WER results are shown in Table 5. The RNNLMs trained by the MTL methods reduced WERs on both sets. The lowest WERs were again achieved by our proposed method of RNNLMs with HMTL, on both the Dev set and the Eval set, where the WER reduction was 0.06% absolute on the Dev set and 0.09% absolute on the Eval set, in comparison with RNNLM without MTL. Due to the casual conversation style, and the very limited training data, the linguistic patterns were hard to learn on the AMI task. The casual text style could also have caused higher errors in the linguistic labels generated automatically by the toolkit algorithms. The unreliable linguistic information would lead to inferior quality of word representations in MTL, making the WER reductions not as effective as on the WSJ task. However, the WER results suggested that the RNNLM still benefited from the additional knowledge of the auxiliary NLP tasks.

	Dev	Eval
Baseline trigram	23.15	23.39
+ RNNLM (w/o MTL)	21.91 ± 0.09	22.20 ± 0.03
+ RNNLM (w/ FMTL)	21.89 ± 0.03	22.14 ± 0.03
+ RNNLM (w/ HMTL)	$\mathbf{21.85 \pm 0.06}$	$\mathbf{22.11 \pm 0.05}$

Table 5: WER (%) results on AMI corpus

An ablation study was also performed on the AMI dataset. The outcome patterns were similar to those observed on the WSJ corpus, with the major WER reduction due to the chunking task, and the other two tasks contributed less to the model performance.

	Dev	Eval
chunking+POS+NER	21.85 ± 0.06	22.11 ± 0.05
chunking+POS	21.89 ± 0.06	22.14 ± 0.05
POS+NER	21.93 ± 0.11	22.20 ± 0.12
chunking+NER	21.90 ± 0.05	22.16 ± 0.05

Table 6: Ablation study of WER (%) results on AMI corpus.

4.2.2. *Perplexities*

The perplexity results are shown in Table 7. The RNNLMs with both MTL methods had higher PPLs in comparison with the RNNLMs without MTL. We argue that this was caused by the casual text style of AMI, which brought about a higher level of label noise than WSJ in the NLP tasks.

	Dev	Eval
RNNLM (w/o HMTL)	54.2 ± 0.6	50.8 ± 0.7
RNNLM (w/ FMTL)	55.0 ± 1.8	52.1 ± 1.0
RNNLM (w/ HMTL)	56.7 ± 4.0	53.6 ± 4.6

Table 7: Perplexity results on AMI corpus

5. CONCLUSIONS AND FUTURE WORK

We have proposed and investigated an approach of hierarchical multi-task learning for RNNLM to improve its generalization performance for speech recognition. We utilized three NLP tasks as auxiliary tasks. These tasks were supervised in a hierarchy in the order of chunking, POS tagging, and named entity recognition. On speech recognition tasks of WSJ and AMI, improved WER results have been achieved by the proposed method, demonstrating the promising potential of our RNNLM with HMTL approach for speech recognition. By supervising NLP tasks in a hierarchical manner, the RNNLMs have learned rich word representations for word prediction. In the future, we plan to investigate other related auxiliary tasks to enhance the RNNLM performance in speech recognition.

6. REFERENCES

[1] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Interspeech*, 2010, pp. 1045–1048.

[2] Jan Niehues, Thanh-Le Ha, Eunah Cho, and Alex Waibel, "Using factored word representation in neural network language models," in *Proceedings of the First Conference on Statistical Machine Translation (WMT 2016)*, 2016, pp. 74–82.

[3] Kaisheng Yao, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu, "Recurrent neural networks for language understanding," in *Interspeech*, 2013, pp. 2524–2528.

[4] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] Martin Sundermeyer, Hermann Ney, and Ralf Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. Audio, Speech and Language Processing*, vol. 23, no. 3, pp. 517–529, 2015.

[6] Chihiro Shibata, Kei Uchiumi, and Daichi Mochihashi, "How LSTM encodes syntax: Exploring context vectors and semi-quantization on natural text," in *COLING*, 2020, pp. 4033–4043.

[7] Andrei Alexandrescu and Katrin Kirchhoff, "Factored neural language models," in *NAACL*, 2006, pp. 1–4.

[8] Tomas Mikolov and Geoffrey Zweig, "Context dependent recurrent neural network language model," in *SLT*, 2012, pp. 234–239.

[9] Youzheng Wu, Xugang Lu, Hitoshi Yamamoto, Shigeki Matsuda, Chiori Hori, and Hideki Kashioka, "Factored language model based on recurrent neural network," in *COLING*, 2012, pp. 2835–2850.

[10] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[11] Rich Caruana, "Multitask learning," *Machine Learning*, vol. 28, pp. 41–75, 1997.

[12] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. 76, pp. 2493–2537, 2011.

[13] Anders Søgaard and Yoav Goldberg, "Deep multi-task learning with low level tasks supervised at lower layers," in *ACL)*, 2016, pp. 231–235.

[14] Victor Sanh, Thomas Wolf, and Sebastian Ruder, "A hierarchical multi-task approach for learning embeddings from semantic tasks," *AAAI*, vol. 33, no. 01, pp. 6949–6956, 2019.

[15] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher, "A joint many-task model: Growing a neural network for multiple NLP tasks," in *EMNLP*, 2017, pp. 1923–1933.

[16] Douglas B. Paul and Janet M. Baker, "The design for the wall street journal-based CSR corpus," in *DARPA Speech and Language Workshop*, 1992, HLT '91, pp. 357–362.

[17] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner, "The AMI meeting corpus: A pre-announcement," *MLMI International Workshop*, 2005.

[18] Minguang Song, Yunxin Zhao, and Shaojun Wang, "Multi-objective multi-task learning on rnnlm for speech recognition," in *SLT*, 2018, pp. 197–203.

[19] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[20] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukás Burget, Ondrej Glembek, Nagendra Kumar Goel, Mirko Hannemann, Petr Motlícek, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[21] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky, "The Stanford CoreNLP natural language processing toolkit," in *ACL*, 2014, pp. 55–60.

[22] Alan Akbik, Duncan Blythe, and Roland Vollgraf, "Contextual string embeddings for sequence labeling," in *COLING*, 2018, pp. 1638–1649.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NIPS*, 2019.

[24] Karel Veselý, Arnab Ghoshal, Lukáš Burget, and Daniel Povey, "Sequence-discriminative training of deep neural networks," in *Interspeech*, 2013, pp. 2345–2349.

[25] Stanley Chen, Douglas Beeferman, and Ronald Rosenfeld, "Evaluation metrics for language models," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[26] Philip Clarkson and Tony Robinson, "Towards improved language model evaluation measures," in *EuroSpeech*, 1999.

[27] Minguang Song, Yunxin Zhao, Shaojun Wang, and Mei Han, "Learning recurrent neural network language models with context-sensitive label smoothing for automatic speech recognition," in *ICASSP*, 2020, pp. 6159–6163.

[28] Minguang Song, Yunxin Zhao, Shaojun Wang, and Mei Han, "Word similarity based label smoothing in RNNLM training for ASR," in *SLT*, 2021, pp. 280–285.

[29] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, "Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI," in *Interspeech*, 2016, pp. 2751–2755.