Interspeech 2022
18-22 September 2022, Incheon, Korea

A BERT-based Language Modeling Framework

Chin-Yueh Chien[12], *and Kuan-Yu Chen*[2]

[1]KenKone Medical Co., Ltd., Taiwan
[2]National Taiwan University of Science and Technology, Taiwan
chris.chien@kenkone.com, kychen@mail.ntust.edu.tw

Abstract

Deep learning has brought considerable changes and created a new paradigm in many research areas, including computer vision, speech processing, and natural language processing. In the context of language modeling, recurrent-based language models and word embedding methods have been pivotal studies in the past decade. Recently, pre-trained language models have attracted widespread attention due to their enormous success in many challenging tasks. However, only a dearth of works concentrates on creating novel language models based on the pre-trained models. In order to bridge the research gap, we take the bidirectional encoder representations from Transformers (BERT) model as an example to explore novel uses of a pre-trained model for language modeling. More formally, this paper proposes a set of BERT-based language models, and a neural-based dynamic adaptation method is also introduced to combine these language models systematically and methodically. We conduct comprehensive studies on three datasets for the perplexity evaluation. Experiments show that the proposed framework achieves 11%, 39%, and 5% relative improvements over the baseline model for Penn Treebank, Wikitext 2, and Tedlium Release 2 corpora, respectively. Besides, when applied to rerank *n*-best lists from a speech recognizer, our framework also yields promising results compared with baseline systems.

Index Terms: pre-trained language model, BERT, language modeling, dynamic adaptation

1. Introduction

Speech, which conveys and expresses thoughts and emotions, is the most direct way for people to communicate with each other. Natural language processing is a field that people have researched for a long time [1-3]. One of the holy grails is to interact with machines naturally. With the proliferation of personal devices and massive increases in computing power, people are now using machines to perform tasks that require at least some level of intelligence, leading to rapid advances in machine learning and related technologies. Recently, deep learning has been the most trending research, a subset of machine learning and highly correlated with neural networks. With the evolution of deep learning, various research fields, such as computer vision [4, 5], speech processing [6, 7], natural language processing [8-10], and so on, have reached a new paradigm.

After the abilities and capabilities of recurrent-based neural networks for language modeling were revealed [11-13], research on learning language representations by inferring low-dimensional dense representations of words has sprung up [14-16]. Since the set of learned word embeddings captures precise syntactic and semantic information of words, it is usually used to accurately determine the relationships between words and as input features for downstream neural networks. More recently, research trends have shifted towards exploring and using pre-trained language models, typically consisting of a stack of Transformers [17-20]. The major innovations are capturing the interactions among tokens using the self-attention mechanism [17] and updating the model parameters with self-supervised objectives [18-20]. One of the epoch-making approaches is Bidirectional Encoder Representations from Transformers (BERT) models, the success of which comes from a framework where pre-training is followed by task-specific fine-tuning [18]. In recent years, BERT and its variants, such as the XLNet [19], the RoBERTa [20], and the OpenAI GPT [21, 22], have created a dominant paradigm in natural language processing.

Although pre-trained language models have led to massive successes in many tasks, the progress of language modeling seems to be hesitant [23-26]. To leverage the strengths of pre-trained language models for language modeling, we take BERT as an example to develop a novel BERT-based language modeling framework in this paper. On the one hand, following the common usage, we employ BERT to generate a comprehensive representation for a partial word sequence. Based on the representations, two language models are introduced by fine-tuning a few layers of neural networks stacked upon BERT. On the other hand, inspired by the principle of the classic cache language model and its variants [27-29], we design novel methods to adapt the language model with a set of memorized information. Besides, we introduce a neural-based dynamic adaptation method to efficiently combine these models. The proposed framework is evaluated in terms of perplexity and word error rates. Compared with some baseline models, a series of experiments demonstrate the utilities and flexibilities of the proposed framework.

2. Related Work

2.1. *N*-gram Models

One of the goals of a language model is to quantify the possibility of a given word sequence $W = \{w_1, w_2, \dots, w_L\}$ occurring in the natural language. According to the chain rule, a common simplification strategy is to decompose $P(W)$ into a series of conditional probabilities, as

$$P(W) = P(w_1)\prod_{i=2}^{L} P(w_i|W_{<i}), \quad (1)$$

where $W_{<i} = \{w_1, w_2, \dots, w_{i-1}\}$ denotes the partial sequence before w_i. In order to accurately estimate or approximate $P(w_i|W_{<i})$, several strategies have been investigated. Among them, the *n*-gram language model, which assumes each word is only conditioned on its previous $n-1$ words (i.e., $P(w_i|W_{<i}) = P(w_i|w_{i-n+1}, \dots, w_{i-1})$), is the most famous and popular method [1, 30]. The conditional probability is usually estimated based on counting statistics. Paired with smoothing technologies, the *n*-gram model achieves good performances in practice. Along with the development of deep learning, the neural network-based *n*-gram language model (NNLM) is introduced and achieves competitive or even better results [31]. The most simple architecture of NNLM consists of two feed-forward neural networks with a softmax activation:

10.21437/Interspeech.2022-11266

$$P_{NNLM}(w_i|W_{<i}) = \text{softmax}\left(\text{FFN}\left(\tilde{h}_{NNLM}^{w_i}\right)\right), \quad (2)$$

$$\tilde{h}_{NNLM}^{w_i} = \text{FFN}\left(\left[h_{w_{i-n+1}}, \cdots, h_{w_{i-1}}\right]\right), \quad (3)$$

where h_{w_t} denotes a vector representation for w_t, which can either be a one-hot vector or a low-dimensional dense vector by any language representation method. In recent decades, the recurrent neural network language model (RNNLM) further boosts performance on standard language modeling benchmarks [11, 12]. RNNLM is usually framed as

$$P_{RNNLM}(w_i|W_{<i}) = \text{softmax}\left(\text{FFN}\left(\tilde{h}_{RNNLM}^{w_i}\right)\right), \quad (4)$$

$$\tilde{h}_{RNNLM}^{w_i} = \text{FFN}\left(\left[\tilde{h}_{RNNLM}^{w_{i-1}}, h_{w_{i-1}}\right]\right). \quad (5)$$

By considering hidden representation generated by the previous iteration (i.e., $\tilde{h}_{RNNLM}^{w_{i-1}}$), RNNLM naturally encodes and plugs long-span information into language modeling than NNLM. In addition to the recurrent neural network, the long short-term memory [13] and the gated recurrent unit [32] are representatives in practice.

2.2. Cache Models

A simple but elegant adaptation method for the language model is cache, whose idea is a word used in the immediate past is much more likely to be used soon [27]. Take a unigram cache model for example, the probability is estimated by:

$$P_{cache}(w_i|W_{<i}) = \frac{c(w_i, W_{<i})}{|W_{<i}|}, \quad (6)$$

where $|W_{<i}|$ is the number of words before w_i and $c(w_i, W_{<i})$ means the frequency of w_i occurring in $W_{<i}$. Obviously, the strategy estimates a domain-aware language model, so proper nouns and/or rare words can receive higher probabilities in specific contexts. Therefore, it can compensate the n-gram model for long-range contextual information. Recently, a neural network-based cache model has been introduced [28]. The central idea is to create a memory-like datastore, which records a set of key-value pairs $\left(h_{W_{<t}}, w_t\right)$. In detail, for each pair of word w_t and its previous words $W_{<t}$, a key generating function is employed to encode $W_{<t}$ to be a dense vector representation $h_{W_{<t}}$, and the value is the word itself. Subsequently, the neural cache language model (NCLM) is formulated as:

$$P_{NCLM}(w_i|W_{<i}) \propto \sum_{t=1}^{i-1} \mathbb{I}_{\{w_t = w_i\}} \exp\left(\theta h_{W_{<t}}^{\top} h_{W_{<i}}\right), \quad (7)$$

where θ is a temperature scaler to control the shape of the probability distribution, and $\mathbb{I}$ denotes an indicator function. After that, a k-NN language model (KNNLM), which can be viewed as a generalization of NCLM, is presented [29]. It creates a datastore and records all the key-value pairs extracted from the training corpus. Comparing NCLM with KNNLM, the latter concentrates on explicitly memorizing training examples to better generalize to similar cases at test time, while the former follows the principle of the classic cache language model and provides a dynamic method to improve domain adaptation.

3. The Proposed Framework

Among popular pre-trained language models, the BERT model [18] has attracted much interest due to its representative and state-of-the-art performances in several subjects. In this paper, we concentrate on exploring novel uses of BERT in the context of language modeling.

3.1. A Cross-layer BERT-based Language Model

The naïve yet efficient way to leverage BERT for a task is by treating BERT as an input encoder and adjusting a set of additional parameters to optimize the performance of the target task. For language modeling, our goal is to predict the most possible word w_i occurring after a given sequence of words $W_{<i} = \{w_1, w_2, \ldots, w_{i-1}\}$. Hence, without loss of generality, to leverage BERT for language modeling, the input is $W_{<i}$. As with the seminal literal, a tokenizer is employed to split $W_{<i}$ into a token sequence $\left\{t_1^{W_{<i}}, \cdots, t_L^{W_{<i}}\right\}$ and special tokens $[CLS]$ and $[SEP]$ are padded at the beginning and the end of the sequence, respectively. Besides, position embeddings are used to distinguish the order of each token in the line. After that, BERT performs a layer-by-layer feature extraction for the token sequence $\{[CLS], t_1^{W_{<i}}, \ldots, t_L^{W_{<i}}, [SEP]\}$:

$$\begin{aligned}&\left[h_{[CLS]}^k; h_1^k; \ldots; h_L^k; h_{[SEP]}^k\right] \\ &= BERT^k\left(\left[h_{[CLS]}^{k-1}; h_1^{k-1}; \ldots; h_L^{k-1}; h_{[SEP]}^{k-1}\right]\right)\end{aligned}, \quad (8)$$

where k is the index of the Transformer layer in BERT and h_l^k denotes the representation generated by k^{th} Transformer for l^{th} token. The latest representation of $[CLS]$, which is designed to encapsulate all the useful information distilled from $W_{<i}$, can be viewed as a comprehensive vector for $W_{<i}$. Based on the representation, a single feed-forward neural network with softmax activation is adopted to translate $h_{[CLS]}^K$ to a probability distribution:

$$P_{BLM}(w_i|W_{<i}) = \text{softmax}\left(\text{FFN}\left(h_{[CLS]}^K\right)\right). \quad (9)$$

We denote the simple BERT-base language model as BLM in short. Here, it is worth noting that the output is a probability distribution over words in the lexicon instead of tokens used in BERT. In contrast to previous studies, which usually can only calculate pseudo-likelihood for a word sequence, the proposed BLM (and all the models presented in this paper) can fairly compare with conventional language models [33-35].

Although BLM provides a straight manner to explore BERT for language modeling, it seems to only use a portion of the merits of BERT. Besides, according to the previous literal, each layer of Transformer in BERT encodes different grains of information of a natural language [36-38]. We thus extend the principle and present a cross-layer BERT-based language model (xBLM), which is designed to plug all the distilled information into a language model. A bit of terminology, the xBLM is formulated as:

$$P_{xBLM}(w_i|W_{<i}) = \text{softmax}\left(\text{FFN}\left(\tilde{h}_{xBLM}\right)\right), \quad (10)$$

$$\tilde{h}_{xBLM} = \sum_{k=1}^{K} \alpha^k h_{[CLS]}^k, \quad (11)$$

where $\tilde{h}_{xBLM}$ is a weighted sum of all $[CLS]$ representations corresponding to every Transformer layer of BERT and α^k, that reveals the importance of each level of information, is determined by a self-attention mechanism and a single feed-forward neural network with softmax activation:

$$[\alpha^1; \ldots; \alpha^K] = \text{softmax}\left(\text{FFN}\left(\text{selfatt}\left(\left[h_{[CLS]}^1; \ldots; h_{[CLS]}^K\right]\right)\right)\right). \quad (12)$$

3.2. BERT-based Cache Models

Inspired by the cache-based language models, we intend to gather advantages from the principle of cache modeling and the pre-trained language models. To crystalize the idea, we first employ BERT to produce vector representations for each sub-sequence of a given partial word sequence. More formally, for a given partial sequence $W_{<i} = \{w_1, w_2, \ldots, w_{i-1}\}$, $i-2$ sub-sequences $\{W_{<2}, W_{<3}, \ldots, W_{<i-1}\}$ can be obtained, and their unique vector representations $\left\{h_{[CLS]}^{K,W_{<2}}, \ldots, h_{[CLS]}^{K,W_{<i-1}}\right\}$ are generated by the last Transformer layer in BERT corresponding to the $[CLS]$ token at the beginning of each sub-sequence. Accordingly, a BERT-based cache language model (cBLM) is constructed as:

$$P_{cBLM}(w_i|W_{<i}) = \text{softmax}\left(\text{FFN}\left(\tilde{h}_{cBLM}^{w_i}\right)\right), \quad (13)$$

$$\tilde{h}_{cBLM}^{w_i} = \sum_{t=2}^{i-1} \alpha^{W_{<t}} h_{[CLS]}^{K,W_{<t}}, \quad (14)$$

where the weighting factor $\alpha^{W_{<t}}$ is calculated by:

$$\alpha^{W_{<t}} \propto \exp\left(-d\left(h_{[CLS]}^{K,W_{<i}}, h_{[CLS]}^{K,W_{<t}}\right)\right), \quad (15)$$

where d denotes a distance measure for a pair of vectors. In short, cBLM uses immediately historical information to form a language model, mimicking the ideas of the conventional cache model and the NCLM. The difference is that both the conventional cache language model and NCLM work at the word level, while cBLM processes the information at the embedding space. This way, cBLM has more flexibility since it benefits from a pre-trained language model and is sensitive to the vicinity information.

A reasonable extension of cBLM is to expand the cache information beyond the immediate vicinity to include the entire training corpus. We, therefore, create a datastore $\mathcal{D}$ that contains vector representations of all sub-sequences collected from the training text. Following, for each partial word sequence $W_{<i}$, we retrieve M nearest neighbors $\{h_{[CLS]}^{K,1}, \dots, h_{[CLS]}^{K,m}, \dots, h_{[CLS]}^{K,M}\}$ from $\mathcal{D}$. The conditional probability $P_{gcBLM}(w_i|W_{<i})$ can thus be computed by referring to the neighbors:

$$P_{gcBLM}(w_i|W_{<i}) = \text{softmax}\left(\text{FFN}(\tilde{h}_{gcBLM}^{w_i})\right), \tag{16}$$

$$\tilde{h}_{gcBLM}^{w_i} = \sum_{m=1}^{M} \alpha^m h_{[CLS]}^{K,m}. \tag{17}$$

Similarly, α^m is a weighting factor in revealing the importance of each neighbor and is determined based on the distance between $h_{[CLS]}^{K,W_{<i}}$ and $h_{[CLS]}^{K,m}$:

$$\alpha^m \propto \exp\left(-d\left(h_{[CLS]}^{K,W_{<i}}, h_{[CLS]}^{K,m}\right)\right), \tag{18}$$

where d denotes a distance measure. We term the model a BERT-based global cache language model (gcBLM). As a counterpart of cBLM, gcBLM memorizes training examples to better generalize to similar cases at test time. However, it should be noted that gcBLM needs extra memory space to keep the extremely large datastore and requires additional computations to retrieve neighbors from the datastore.

3.3. A Neural-based Dynamic Adaptation Method

After presenting novel uses of BERT for language modeling, we intend to combine these language models dynamically and systematically at the last step. For a partial sequence $W_{<i}$, intermediate representations $\tilde{h}_{xBLM}$, $\tilde{h}_{cBLM}^{w_i}$, and $\tilde{h}_{gcBLM}^{w_i}$ for xBLM, cBLM, and gcBLM can be inferred, respectively. NNLM and recurrent-based language model implemented with long short-term memory (LSTMLM) are also combined to account for variations in short- and long-term word regularities. Their intermediate representations are $\tilde{h}_{NNLM}^{w_i}$ and $\tilde{h}_{LSTMLM}^{w_i}$ (cf. Section 2.1). We realize a dynamic adaptation function based on these embeddings using a self-attention mechanism and a feed-forward neural network with softmax activation:

$$[\alpha_{NNLM}; \alpha_{LSTMLM}; \alpha_{xBLM}; \alpha_{cBLM}; \alpha_{gcBLM}] = \text{softmax}\left(\text{FFN}\left(\text{selfatt}([\tilde{h}_{NNLM}^{w_i}; \tilde{h}_{LSTMLM}^{w_i}; \tilde{h}_{xBLM}^{w_i}; \tilde{h}_{cBLM}^{w_i}; \tilde{h}_{gcBLM}^{w_i}])\right)\right). \tag{19}$$

This architecture allows each vector to communicate with others and decides the importance of each language model. Finally, a resulting mixture language model can be obtained:

$$P_{mix}(w_i|W_{<i}) = \text{softmax}\left(\text{FFN}(\tilde{h}_{mix}^{w_i})\right), \tag{20}$$

$$\tilde{h}_{mix}^{w_i} = \alpha_{NNLM}\tilde{h}_{NNLM}^{w_i} + \alpha_{LSTMLM}\tilde{h}_{LSTMLM}^{w_i} + \alpha_{xBLM}\tilde{h}_{xBLM}^{w_i} + \alpha_{cBLM}\tilde{h}_{cBLM}^{w_i} + \alpha_{gcBLM}\tilde{h}_{gcBLM}^{w_i}. \tag{21}$$

Despite the conventional linear interpolation method, the neural-based dynamic adaptation method provides an alternative way to dynamically hybrid a set of neural-based language models.

4. Experiments

4.1. Experimental Setup

To examine the efficiency of the proposed BERT-based language modeling framework, we report log perplexity per word for the language modeling task and word error rate for the task of reranking n-best lists from a speech recognizer. For the perplexity evaluation, experiments are conducted on Penn Treebank [39], Wikitext 2 [40], and Tedium Release 2 [41] corpora. We train language models on the training set, select hyper-parameters based on the performance of the development set, and use the resulting model to evaluate the test set. The lexicon sizes are 10,000, 33,278, and 159,848 for Penn Treebank, Wikitext 2, and Tedium Release 2, respectively. For the n-best reranking task, our speech recognition system is built up using the Kaldi toolkit based on the TDNN-LSTM recipe [42, 43]. In audio processing, spectral analysis is applied to a 25 ms frame of speech waveform every 10 ms. 40 MFCCs derived from 40 FBANKs, plus 3 pitch features, are used for each acoustic frame. Utterance-based mean subtraction is applied to these features. For each speech utterance, 100 candidates are generated by the speech recognizer.

The training procedure of our framework is as follows. First, based on the pre-trained BERT model (BERT-base-uncased), we fine-tune the set of additional parameters toward minimizing the negative log-likelihood of the training corpus for BLM. After that, xBLM is obtained by updating only the additional self-attention mechanism and feed-forward neural networks. Then, we use xBLM to extract comprehensive representations for word sequences, and the FAISS toolkit [44] is used to compute the distance between two vectors for building cBLM and gcBLM. Meanwhile, it is worth noting that cBLM and gcBLM adopt additional information to adapt the language models by weighted averaging comprehensive vectors without extra training and model fine-tuning. Finally, the parameters, including the self-attention mechanism and feed-forward neural network, of the dynamic adaptation method are updated to minimize the negative log-likelihood of the training corpus. In other words, we freeze the component language models and only fine-tune the parameters of the dynamic adaptation method for each combination. For the proposed framework, the hidden layer size is set to 768, and Adam [45] is selected as the optimizer. For BLM, xBLM, and the neural-based dynamic adaptation method, learning rates are all set to 0.0001, and batch sizes are 32, 512, and 256, respectively. The hyperparameter M for gcBLM is empirically set to 128 without any tunning.

4.2. Experimental Results

4.2.1. Perplexity

Table 1 summarizes all the experimental results for the perplexity evaluation concerning different models, settings, and corpora. The hollow circle "。" means the combination is by the proposed neural-based dynamic adaptation method, whereas the plus symbol "+" denotes the linear interpolation method. Several interesting observations and worthwhile discussions can be drawn. First of all, we can find neural network-based tri-gram model (NNLM) outperforms the conventional tri-gram language model (Trigram) by a large margin. When combined with LSTMLM, NNLM。LSTMLM at least achieves absolute improvements of 17 perplexity units compared to Trigram+LSTMLM in all cases. Next, when combined with xBLM, NNLM。LSTMLM。xBLM provides superior improvements over NNLM。LSTMLM in all cases because the information from BERT is exploited. If we further fuse all the models together, NNLM。LSTMLM。xBLM。cBLM。gcBLM shows about 11%, 39%, and 5% relative improvements over Trigram+LSTMLM+xBLM+cBLM+gcBLM for Penn

Table 1: *The perplexity and word error rates with respect to different models, settings and datasets.*

	Perplexity						Word Error Rate	
	Penn Treebank		Wikitext 2		Tedlium Release 2		Tedlium Release 2	
Model	Dev.	Test	Dev.	Test	Dev.	Test	Dev.	Test
Trigram	229.86	220.93	274.15	253.32	213.32	195.11	8.12	8.23
Trigram + LSTMLM	100.18	98.23	130.63	127.51	134.81	137.28	7.66	7.72
Trigram + LSTMLM + xBLM + cBLM + gcBLM	53.46	52.81	52.03	51.58	56.00	56.61	6.49	6.69
NNLM	182.86	174.64	207.30	210.66	169.12	175.08	8.02	8.12
NNLM ◦ LSTMLM	82.36	79.61	107.15	105.28	110.26	113.01	7.31	7.45
NNLM ◦ LSTMLM ◦ xBLM	48.97	48.64	33.99	33.51	55.16	55.45	6.46	6.61
NNLM ◦ LSTMLM ◦ xBLM ◦ cBLM ◦ gcBLM	**47.56**	**46.87**	**31.41**	**31.01**	**53.05**	**53.77**	**6.38**	**6.51**
NNLM ◦ BLM	56.84	56.46	54.76	53.84	63.53	64.04	6.75	6.95
NNLM ◦ xBLM	53.11	52.78	37.86	37.03	57.00	57.57	6.58	6.77
NNLM ◦ cBLM	65.99	65.64	68.89	68.01	79.49	81.83	7.10	7.22
NNLM ◦ gcBLM	66.31	65.16	72.74	71.89	81.51	83.91	7.10	7.22
Dual mdLSTM [46]	46.91	46.09	53.48	50.71				
FRAGE + AWD-LSTM-MoS [47]	47.38	46.54	40.85	39.14				
Adversarial + AWD-LSTM-MoS[48]	46.63	46.01	40.27	38.65				
BERT-Large-CAS [49]	**36.14**	**31.34**	**37.79**	34.11				
GPT-2 [21]		35.76		**18.34**				
TDNN + RNNLM Rescoring							8.60	8.90
Transformer + LSTMLM Rescoring							9.30	8.10

Treebank, Wikitext 2, and Tedlium Release 2 corpora, respectively. The set of experiments not only validates the feasibility of the neural-based language models, but also reveals the potential of the proposed neural-based dynamic adaptation method.

Next, we turn to examine each proposed BERT-based model. At first glance, BERT-based models all consistently outperform LSTMLM, which may be because they all receive brilliant benefits from BERT. As expected, xBLM gives better results than BLM since it takes different grains of information into account, while BLM is considered an acceptable model in practice. Although cBLM and gcBLM bring similar results, the former is slightly better than the latter. The results signal that contextually historical information seems more critical than information extracted from training data. Another notable observation is that cBLM and gcBLM are both worse than xBLM and BLM. The reason is that the most critical vector $h_{[CLS]}^{\lambda,w_{<t}}$ is only used for selecting candidates and calculating distances, not for constructing the resulting language models. Accordingly, one of our emergent future works is to reformulate cBLM and gcBLM by carefully using the critical vectors. In sum, cBLM and gcBLM are suitable for compensating xBLM and BLM, and their performance is inherently competitive.

Finally, some strong baselines are compared with the proposed framework, including Dual mdLSTM [46], FRAGE+AWD-LSTM-MoS [47], Adversarial+AWD-LSTM-MoS [48], BERT-Large-CAS [49], and GPT-2 [21]. From Table 1, the proposed framework yields competitive and/or even better results than these baseline models. Here, it should be mentioned that our model size is 190M, while BERT-Large-CAS and GPT-2 are 395M and 1,542M, respectively. Thus, the experiments demonstrate both the efficiency and effectiveness of the proposed framework.

4.2.2. N-Best Reranking

In the set of experiments, we apply the proposed framework to rerank *n*-best lists from a speech recognizer. All the results are also listed in Table 1. Looking over the table, the performance trends are aligned well with the perplexity evaluation task. On the one hand, the BERT-based models can outperform conventional models (i.e., Trigram, NNLM, and LSTMLM) by a large margin because they inherit the capabilities and abilities from BERT. Compared with NNLM and NNLM ◦ LSTMLM, the proposed hybrid model achieves over 20% and 12% relative improvements on the test set, respectively. On the other hand, the neural-based dynamic adaptation method provides a systematic and methodical way to dynamically hybrid neural-based language models so as to achieve remarkable results than the conventional linear interpolation method. Consequently, the proposed neural-based dynamic adaptation method obtains 2.7% relative improvements on the test set for combining all the language models compared to the traditional linear interpolation method. In the end, we compare the proposed framework to two classic baselines, including the TDNN model with RNNLM rescoring [43] and the attention-based encoder-decoder model with LSTMLM rescoring [50]. From the results, we can conclude that the proposed framework is deemed the superior vehicle for the *n*-best reranking task as expected.

5. Conclusion

This study presented a BERT-based language modeling framework, including a set of novel language models and a neural-based dynamic adaptation method. The proposed framework has been evaluated on perplexity and *n*-best reranking tasks. A series of experiments demonstrates remarkable superiority over other baselines, thereby indicating the potential of the framework. We believe the framework can be generalized to other pre-trained language models, and we leave the extension for future investigations. In the future, we also plan to evaluate the proposed framework in different languages and extend it to other natural language processing-related tasks, such as spoken document retrieval and spoken visual question answering.

6. Acknowledgments

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 111-2636-E-011-005 through the Young Scholar Fellowship Program. We thank the National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational and storage resources.

7. References

[1] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing.* Computational Linguistics, and Speech Recognition. Upper Saddle River, N.J.: Prentice Hall, 2000.

[2] M. Benzeghiba et al., "Automatic speech recognition and speech variability: A review," *Speech Communication*, vol. 49, Issues 10–11, pp. 763–786, 2007.

[3] R. Baeza-Yates and B. Ribeiro-Neto, "Modern information retrieval," in ACM press New York, 1999.

[4] A. Bochkovskiy et al., "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

[5] P. Zhang et al., "VinVL: Revisiting Visual Representations in Vision-Language Models," *arXiv preprint arXiv:2101.00529*, 2021.

[6] A. B. Nassif et al., "Speech Recognition Using Deep Neural Networks: A Systematic Review," *IEEE Access*, vol. 7, pp. 19143–19165, 2019.

[7] F. H. Yu et al., "Non-Autoregressive ASR Modeling Using Pre-Trained Language Models for Chinese Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1474–1482, 2022.

[8] Y. Liu, "Fine-tune BERT for extractive summarization," *arXiv preprint arXiv:1903.10318*, 2019.

[9] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in *Proceedings of EMNLP*, pp. 6769–6781, 2020.

[10] K. Irie et al., "Language modeling with deep Transformers," in *Proceedings of INTERSPEECH*, 2019.

[11] M. Tomáš et al., "Recurrent neural network based language model," in *Proceedings of INTERSPEECH*, 2010.

[12] T. Mikolov et al., "Extensions of recurrent neural network language model," in *Proceedings of ICASSP*, pp. 5528–5531, 2011.

[13] M. Sundermeyer et al., "LSTM neural networks for language modeling," in *Proceedings of INTERSPEECH*, 2012.

[14] T. Mikolov et al., "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[15] J. Pennington et al., "Glove: Global vectors for word representation," in *Proceedings of EMNLP*, pp. 1532–1543, 2014.

[16] M. Peters et al., "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[17] A. N. Gomez et al., "Attention is all you need," in *Proceedings of NIPS*, pp. 5998–6008, 2017.

[18] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

[19] Z. Yang et al., "XLNet: Generalized autoregressive pretraining for language understanding," in *Proceedings of NIPS*, pp. 5754–5764, 2019.

[20] Y. Liu et al., "RoBERTa: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[21] A. Radford et al., "Language Models are Unsupervised Multitask Learners," 2019.

[22] T. B. Brown et al., "Language Models are Few-Shot Learners," *arXiv preprint arXiv:2005.14165*, 2020.

[23] L. Dong et al., "Unified language model pre-training for natural language understanding and generation." in *Proceedings of NIPS*, pp. 13063–13075, 2019.

[24] K. Irie et al., "RADMM: recurrent adaptive mixture model with applications to domain robust language modeling," in *Proceedings of ICASSP*, pp. 6079–6083, 2018.

[25] M. Hentschel et al., "A unified framework for feature-based domain adaptation of neural network language models," in *Proceedings of ICASSP*, pp. 7250–7254, 2019.

[26] K. Li et al., "An Empirical Study of Transformer-Based Neural Language Model Adaptation," in *Proceedings of ICASSP*, pp. 7934–7938, 2020.

[27] R. Kuhn et al., "A cache-based natural language model for speech recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 6, pp. 570–583, 1990.

[28] E. Graveet al., "Improving neural language models with a continuous cache," in *Proceedings of ICLR*, 2017.

[29] U. Khandelwal et al., "Generalization through memorization: Nearest neighbor language models," in *Proceedings of ICLR*, 2020.

[30] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of ACL*, pp. 310–318, 1996.

[31] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[32] K. Cho et al., "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[33] J. Salazar et al., "Masked language model scoring," in *Proceedings of ACL*, pp. 2699–2712, 2020.

[34] J. Shin et al., "Effective sentence scoring method using BERT for speech recognition," in *Proceedings of ACML*, pp. 1081–1093, 2019.

[35] J. Shin et al., "Fast and Accurate Deep Bidirectional Language Representations for Unsupervised Learning," in *Proceedings of ACL*, pp. 823–835, 2020.

[36] B. Aken et al., "How Does BERT Answer Questions? A Layer-Wise Analysis of Transformer Representations," in *Proceedings of CIKM*, pp 1823–1832, 2019.

[37] W. Vries et al., "What's so special about BERT's layers? A closer look at the NLP pipeline in monolingual and multilingual models," in *Proceedings of EMNLP*, 2020.

[38] Joseph F. DeRose et al., "Attention Flows: Analyzing and Comparing Attention Mechanisms in Language Models," *IEEE Trans. Vis. Comput. Graph*, 27(2): 1160–1170, 2021.

[39] M. Marcus et al., "Building a large annotated corpus of English: The Penn Treebank," 1993.

[40] S. Merity et al., "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[41] A. Rousseau et al., "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *Proceedings of ICLR*, pp. 3935–3939, 2014.

[42] V. Peddinti et al., "A time delay neural network architecture for efficient modeling of long temporal contexts.," in *Proceedings of INTERSPEECH*, 2015.

[43] D. Povey et al., "The Kaldi speech recognition toolkit," in *Proceedings of ASRU*, 2011.

[44] J. Johnson et al., "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, 2019.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of ICLR*, 2014.

[46] C. Oliva, and L. F. Lago-Fernández, "The Importance of the Current Input in Sequence Modeling," *arXiv preprint arXiv:2112.11776*, 2021.

[47] C. Gong et al., "Frage: Frequency-agnostic word representation," in *Proceedings of NIPS*, 2018.

[48] D. Wang et al., "Improving neural language modeling via adversarial training," in *Proceeding of ICML*, 2019.

[49] C. Wang et al., "Language models with transformers," *arXiv preprint arXiv:1904.09408*, 2019.

[50] S. Watanabe et al., "Espnet: End-to-end speech processing toolkit," in *Proceedings of INTERSPEECH*, 2018.