Incorporating Morphological Compositions with Transformer to Improve BERT

Yuncheng Song¹, Shuaifei Song², Juncheng Ge³, Menghan Zhang⁴ and Wei Yang¹*

¹Soft engineering, University of Science and Technology of China, Hefei, Anhui Province, 230000, China
²Soft engineering, University of Science and Technology of China, Hefei, Anhui Province, 230000, China
³Soft engineering, University of Science and Technology of China, Hefei, Anhui Province, 230000, China
⁴Soft engineering, University of Science and Technology of China, Hefei, Anhui Province, 230000, China
*Corresponding author’s e-mail: qubit@ustc.edu.cn

Abstract. BERT model achieves huge performance gains by modeling words and their subwords as input units. However, it still neglects the semantic information of morpheme which has been verified in many previous works. In this paper, we propose Transformer Morpheme Model (TMM), which is based on BERT and explores the effect of morpheme. Since the process of previous works about morpheme are context-independent, TMM model adopts Transformer to process morpheme information on the input layer to overcome this problem. Experiments on MRPC task are conducted to validate the feasibility of our model. TMM model has achieved about 1% gains over BERT model on MRPC task. The results demonstrate the superiority of our method and the effectiveness of morpheme information in the BERT model.

1. Introduction
As a alphabetic script, every letter in English is evolved from a picture which depicting an animal or shape of object. However, we could see from Table 1 that English letters disable to express semantic information and English morpheme is the smallest carrier with semantic, which is different from Chinese. The existence of morpheme is not inly decorative but also functional. In fact, the effectiveness of probing the English morphemes has been validated by many previous works. In word embedding, some works learn higher quality of embeddings by directly encoding the representations of morpheme.

Table 1. Comparison between English and Chinese.

| English    | Chinese   |
|------------|-----------|
| Character  |目，艹，犭…|
| Word       |眼，花，狗…|
| Phrase     |狮子，眼花缭乱…|
| Sentence   |我们做了大量的研究|
to context words or optimizes a joint objective over morphological properties and distributional statistics([1];[2];[3];[4]). While others employ probabilistic graphical regulation to capture the relationship of words and their morphological compositions([5];[6]). Although those above-mentioned researches pay main attention to morphemes, they only utilize the morphological information on embeddings. They obtain a high-quality embeddings by incorporating morphemes, and only evaluate the performance of their embeddings on downstream tasks. These tasks all regard the tasks of Natural Language Processing (NLP) as two independent parts—pre-training embeddings and testing on downstream tasks.

To solve the problem of transfer learning, [7] propose BERT model which surpass humans on two metrics and achieve the best performance in eleven different NLP tests. They add a simple Multi-Layer Perceptron (MLP) or linear classifier to accomplish the downstream tasks after obtaining the word embeddings of BERT. However, BERT only considers the subword unit when modeling at the input layer. It ignores the semantic information of morpheme which has been verified the effectiveness in many previous works.

To explore the effect of morpheme information in the entire NLP tasks, in this paper, we put forward a novel strategy which expands morphemes on BERT model. Since most of the previous studies are context-independent (i.e., adding morphemes directly or encoding semantic information with neural network), our work propose a context-dependent model, named Transformer Morpheme Model (TMM), which modifies the input representation of BERT through incorporating Transformer to process the information of morphemes. For comparison, our model with several state-of-the-art baselines are evaluated on two basic downstream tasks. The results demonstrate that TMM outperform all the baselines and achieve satisfactory improvement on test tasks. In short, the main contributions of this paper are summarized as follows.

- Rather than only incorporating the morphemes in pre-training embeddings, we decide to employ the effect of morphemes on language models which transferring downstream tasks to pre-training embeddings.
- Since the most previous works about morphemes are context-independent, we propose a context-dependent model, named TMM, which expands morphemes on BERT model with Transformer.
- We conduct experiments with several baselines to demonstrate the effectiveness of our strategy. The results show that TMM outperforms the baselines and achieves about 1% gains over BERT model on MRPC task.

2. Related Work

In this section, we first review some related works on morphology-based word embeddings, and then present some language models about transferring pre-training embeddings to downstream tasks.

**Morphology-based Word Embedding** There are many researches propose fine-grained embedding models through incorporating the morphemes of words, e.g., suffixs and affixes. In fact, these reseraches could be divided into two categories.

One category directly encodes the representations of morphological compositions to word embeddings([1];[2];[3];[4];[8]). [1] used Morfessor[9] as morpheme segements and built morphemes tree to learn better embeddings by the recursive neural network. [2] proposed a neural network model to infer the representations of rare or unknown words based on their morphological structure. [3] presented a dynamic strategy for adding representations of morpheme into a vector-based probabilistic language model.[8] could learn character-level semantic information through combining the convolutional character information into words. Although this model captured the character-level information, it costed too many time due to its size. Luckily, [4] construct a log-linear model to bring words, which share similar morphemes, gather together in vector space.

The other category captures the relationship between words and their morphological compositions by probabilistic graph. For example, [5] regard embeddings as latent variables for prior distribution and encode it into neural sequence model to get final result. Moreover, [6] propose a
Gaussian graphical model that is good for smoothing the noisy embeddings through restricting the morphological analysis to pre-trained embeddings. It should be noted that these two models alleviate the problem of predicting unseen words on various degrees. However, these morphology-based models only exploit the morpheme information on embeddings.

**Language Model** Traditionally, the embedding models is a tool that converts the abstracted text of real-world into vectors. These vectors can be operated by mathematical formulas, and the manipulation of vectors is the real tasks in NLP. From the original embedding models to the latest language models([10];[11];[12];[7]), the main contribution is that gradually move the specific downstream tasks to pre-training embeddings.[10]proposed a high efficiency algorithm, named Word2Vec, which learned word embeddings of high-dimensional space can fully reflect the relationship between tokens in the real-world. However, the shortcoming of Word2Vec is that it’s unable to solve the polysemes, since one word is only represented by one embeddings. Fortunately, [11] presented ELMo to solved this problem. They firstly utilized a model to pretrain word embeddings, and the word embedding would be adjusted according to the semantics of context when this word was used in downstream task. The essence of ELMo is that incorporating bi-directional LSTM to dynamically adjust the word embeddings through the current context. After that, [12] adopted Transformer to extract features and obtained further advancement on downstream tasks when compared with ELMo. But they only employed single language model which is single directional Transformer. To this end, [7] proposed BERT model which applied bi-directional Transformer and fine-tuning strategy. The emergence of BERT model has completely changed the relationship between pre-training word embeddings and downstream tasks.

Although BERT model has been achieved milestone success in NLP, it still ignores the semantic information of morphemes which has been proved the effectiveness. In this paper, we employ the English morphemes to provide insights for enhancing the performance of downstream task. Furthermore, since most of morphology-based embedding models are context-independent, we adopt Transformer to process the morphemes information.

3. **Methodology**

We adopt Transformer to modify the input layer of BERT when incorporating the semantic information of morphemes. One specific model, named Transformer Morpheme Model (TMM), are proposed. It should be stated that, although the core of our work is to explore the effect of morpheme on language model, we have found that existing researches about morphemes are context-independent. 
To this end, we use Transformer to solve this defect due to the characteristic of Transformer itself. Figure 1 provides a graphical illustration about our TMM model. Given a token $t_i$ of one sentence, we can obtain the set $s$ of its morphemes from using Morphessor tool (i.e., $t_i$ represents incredible, $s_i$ indicates in, cred and ible in Figure 1). After that, we need convert object into vector. The acquisition of word embeddings $e_w$ is relatively simple, we can get them directly from the pre-training embeddings of BERT. Obtaining the morpheme embeddings is more complicated, we need to modify CBOV model to learn all morpheme embeddings and save them as lookup embedding dictionary. Finally, the morpheme embeddings matrix $e_{si}$ are got through inputing corresponding morphemes into lookup embedding dictionary.

Transformer is the first model built with pure attention and proposed by. It not only has faster calculation speed, but also has outstanding capabilities on extracting semantic feature and capturing long-distance feature. So we employ it to process morpheme information. For word embedding $e_w$ and morpheme embedding matrix $e_{si}$, we use linear mapping to convert $e_w$ to query, $e_{si}$ to key and value. Then we adopt Scaled dot-product attention to calculate query, key and value as following:

$$head = \text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k})V$$

where $Q$, $K$, $V$ represents the query, key, value respectively. $d_k$ shows the dimension of embeddings and $heads$ is one of the multi-head attention in Transformer. Multi-head attention means that it should project $e_w$, $e_{si}$ through $h$ different linear transformations, and finally the results of $h$ head
calculations are concatenate together (h indicates the number of head in multi-head attention). The output of multi-head attention is calculated as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)$$

where $$\text{head}_i = \text{Attention}(QW^Q_i, KW^K_i, VW^V_i) (i \in [1, h])$$

where $$Q$$ is $$e_b$$, $$K$$ and $$V$$ are $$e_b$$. $$\text{Attention}$$ represents the Scaled dot-product attention and Concat means the operation of concatenation. After obtaining the output $$O$$ of multi-head attention, we need add and normalize it with the input $$e_b$$, then encode the result into a connected layer to calculate the final new token embedding $$E_b$$. The formula of connected layer is as follows:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

where $$x$$ is the result of normalization. $$W_1, W_2, b_1, b_2$$ are all parameters of connected layer. Finally, we replace the original token embedding $$e_b$$ of BERT model with the new token embedding $$E_b$$.

**4. Experiments**

In this section, some experimental settings are introduced before conducting experiments.

**4.1 Dataset**

To suit the problems studied in our work, we select Microsoft Research Paraphrase Corpus (MRPC) task as our evaluation dataset. MRPC task is provided by the General Language Understanding Evaluation (GLUE) benchmark [13]. Its main function is to judge whether a pair of sentences form the same new comments are semantically identical. It contains 3669 items for training, 1726 items for testing and only 409 items for deving. In order to retain the raw information of dataset and validate the effectiveness of our method, we preserve all items of MRPC dataset to compare our model with baselines.

**4.2 Experimental Setting**

**Embedding Setting**

The training of BERT model requires powerful computing resources, we are unable to pre-train BERT model due to the limitations of experimental equipment and funding. Unfortunately, the morpheme embeddings are not included in the public vocab which released by BERT pre-training. To solve this problem, we download an English corpus and modified the word2vec tool to train morpheme embeddings. Specifically, we build a morpheme vocab (derived from the resources provided by Michigan State University) and add it into word2vec's substantive vocab. This vocab trains like the vocab which word2vec owns. Finally, we save it in the embedding file and obtain the morpheme embeddings. The dimension of morpheme embeddings is 768 and other training parameters are the same as the original settings of CBOW model.

![Figure 1. The detail introduction of TMM model.](image-url)
Training Setting

We collect all the words from MPRC task and build vocab. For each item in vocab, we adopt Morfessor segmentation tool to split one word into prefix, root and suffix. After that, we construct the relationship between words and their corresponding morphemes by save them into map file (one item in map: incredible*incredible*ible). We add this map into BERT and modify the input layer to obtain our TMM model. The parameters of TMM model are roughly the same as those of BERT-BASE which we base on. The dimension of word embeddings is 768, which is the reason that we set the dimension of morpheme embeddings to 768. In addition, we try some learning rates (1e-5, 2e-5, 3e-5) and finally set it to 2e-5. As for other parameters, we set batch_size to 32, epochs to 3.0 and max_seq_length to 128. The detailed parameters of TMM model are reported in Table 2.

| Parameter              | Value |
|------------------------|-------|
| learning rate          | 2e-5  |
| batch_size             | 32    |
| epochs                 | 3     |
| num_attention_heads    | 12    |
| initializer_range      | 0.02  |

4.3 Baseline Methods

To evaluate the performance of our TMM model, we conduct comparative experiments with the following baselines:

ELMo: ELMo [11] is regarded as one of the state-of-the-art language model. It incorporates Bilstm network with attention mechanism to capture the features. Moreover, it solves the problem of polysemy.

GPT: Unlike ELMo, GPT [12] uses Transformer to extract features and is single directional Transformer. It predicts a word based on the context.

BERT: We choose BERT-BASE [7] as a baseline model since we modify it to obtain our model. Compared with it, TMM model is exactly a further improvement which reasonableness takes the morphemes information into account.

5. Experimental Results

To verify the validity and robustness of our TMM model, we conduct 5 experiments on all models including baseline methods. For the sake of brevity, we are not show all the experimental results. The average results of 5 experiments are listed in Table 3.

| Model | ELMo  | GPT   | BERT  | TMM   |
|-------|-------|-------|-------|-------|
| MRPC  | 84.9  | 82.3  | 85.6  | 86.4  |

Since the three models (ELMo, GPT, BERT) are presented in order, their performance on MRPC task is also progressive, which can be observed in Table 3. Unfortunately, although we have tried many times, we fail to achieve the best performance of BERT-BASE as mentioned in [7]. However, from the readme file of public BERT resource, we find that the result of BERT-BASE on MRPC task is a range of 84%–89%. Any value in the range is reasonable, and the actual performance of BERT-BASE depends on the result of your actual running. Although the result of our actual running about BERT-BASE is not the highest score mentioned in the paper, it is also available value. Besides, we notice that our TMM model surpass all the baseline models. Comparing to the BERT-BASE which our model is mostly based on, TMM model achieves nearly 1% gain over BERT-BASE model on MRPC task. Moreover, it can be found that incorporating morphological compositions with Transformer can contribute to enhancing the performance of BERT model in the downstream tasks.
6. Conclusions
In this paper, we present an explorative study about morpheme information on language model with a special focus on the utilization of Transformer. During the exploration, we found that language model like BERT has achieved excellent performance on downstream tasks without taking the morphemes information into account. At the same time, we also notice that the previous works about morphemes are context-independent. Inspired by these discoveries, we propose our Transformer Morpheme Model (TMM). The experimental results demonstrate the effectiveness and feasibility of our model. In the future, we plan to explore the characteristics of morpheme in greater depth.

References
[1] Luong, T., Socher, R., & Manning, C. (2013, August). Better word representations with recursive neural networks for morphology. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning (pp. 104-113).
[2] Qiu, S., Cui, Q., Bian, J., Gao, B., & Liu, T. Y. (2014, August). Co-learning of word representations and morpheme representations. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers (pp. 141-150).
[3] Botha, J., & Blunsom, P. (2014, January). Compositional morphology for word representations and language modelling. In International Conference on Machine Learning (pp. 1899-1907).
[4] Cotterell, R., & Schütze, H. (2019). Morphological word embeddings. arXiv preprint arXiv:1907.02423.
[5] Bhatia, P., Guthrie, R., & Eisenstein, J. (2016). Morphological priors for probabilistic neural word embeddings. arXiv preprint arXiv:1608.01056.
[6] Cotterell, R., Schütze, H., & Eisner, J. (2016, August). Morphological smoothing and extrapolation of word embeddings. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1651-1660).
[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[8] Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. (2016, March). Character-aware neural language models. In Thirtieth AAAI Conference on Artificial Intelligence.
[9] Creutz, M., & Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. ACM Transactions on Speech and Language Processing (TSLP), 4(1), 3.
[10] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
[11] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
[12] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper. pdf.
[13] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461.
[14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).