Construction of Japanese BERT with Fixed Token Embeddings

Arata Suganami , Hiroyuki Shinnou
Ibaraki University, Department of Computer and Information Sciences
4-12-1 Nakanarusawa, Hitachi, Ibaraki, Japan 316-8511
{22nm721n, hiroyuki.shinnou.0828}@vc.ibaraki.ac.jp

Abstract

In this study, we propose a method to construct Japanese BERT using fixed Token Embedding to reduce BERT model construction time. This method constructs word embeddings in advance using word2vec and the corpus, which is used in learning BERT models. The token embeddings in the BERT model are set to the constructed word embeddings. The parameters of token embeddings are fixed, that is, not learned during BERT model learning. Therefore, the proposed method reduces the processing time of one epoch in BERT model training and obtains the BERT model in fewer epochs. In the experiments, we construct 1024-dimensional 4-layer and 128-dimensional 24-layer Japanese BERT model using conventional and proposed methods, respectively, and the effectiveness of the proposed method was verified.

1 Introduction

BERT (Devlin et al., 2019) is a high-performance pre-trained model, but its large model size requires significant time and computational resources to build. Another problem is the increase in model construction time and cost because of the huge data sets and models that come with the higher performance of pre-trained models (Sharir et al., 2020). To reduce the construction time of BERT, we propose a method to construct Japanese BERT by fixing the word distribution representation. Specifically, we constructed Japanese BERT by learning word embeddings in advance using word2vec (Mikolov et al., 2013) and fixing them as the Token Embedding of BERT. This reduces the time required to learn word embeddings during BERT construction.

In the experiments, we constructed 128-dimensional 24-layer and 1024-dimensional 4-layer Japanese BERTs using conventional and proposed methods, respectively, and the effectiveness of the proposed method was verified by comparing the model construction time and accuracy in a document classification task for Japanese news articles. The results showed that the proposed method reduced the construction time for both BERTs. The results of the document classification task showed that the proposed method can learn a model with the same level of accuracy using fewer epochs.

2 Related Work

2.1 Small size BERT

BERT is a huge model, and its training requires a significant amount of computation time and computer resources. The amount of time required to process inference using BERT is also significant. To address this problem, many studies have been conducted to miniaturize BERT.

DistilBERT (Sanh et al., 2019) is a miniaturized BERT built using distillation (Hinton et al., 2019).

ALBERT (Lan et al., 2019) reduced the number of parameters of BERT using two methods, such as Cross-Layer Parameter Sharing and Factorized Embedding Parameterization. Cross-Layer Parameter Sharing is a method to used for sharing the parameters of each layer of BERT. Factorized Embedding Parameterization decomposes a matrix of size V×H representing the word embedding into a product of a matrix of size V×E and a matrix of size E×H, where E is a small number (e.g. 128), the number of parameters in BERT's word embedding representation can be reduced by approximately 13%.

TinyBERT (Jiao et al., 2020) is a miniaturized version of BERT that uses a two-step distillation process. The first distillation stage is specialized for transformers. The transformer is classified into four, and a loss function is set for each layer. The BERT fine-tuned in the downstream task is used as the teacher model, in the second stage of distillation, while the TinyBERT constructed in the first stage is used as the student model. To perform Transformer-specific distillation, the training data is augmented with task-specific data.

Q8BERT (Zafrir et al., 2019) is a miniaturized version of BERT that quantized the weights of all coupling and embedding layers to 8-bits. However, operations that require high accuracy (softmax and layer normalization) are still 32-bit. In learning, the forward process uses the quantized values of the parameters, while the back backpropagation process uses the pre-quantized values, a technique known as Fake Quantization.

MobileBERT (Sun et al., 2020) is a small and lightweight BERT that can run on small devices. While most miniaturized BERTs are miniaturized by reducing the number of layers, MobileBERT is miniaturized by reducing the dimensionality of the embedding representation. Therefore, BERT-large, which has more layers and a higher dimensionality of the embedding representation than BERT, is targeted for miniaturization, and a Bottleneck layer and an inverse Bottleneck layer are added to the model to match the dimensionality of the embedding representation. The model is trained using Progressive Knowledge Transfer, a method in which training is performed sequentially, starting from the bottom layer.

Poor Man's BERT (Sajjad et al., 2020) is a simple BERT with some layers removed. Fine-tuning can achieve a certain level of performance by simply adjusting the layers to be removed according to the task.

The method proposed in this study does not learn BERT token embeddings. Therefore, the number of parameters to be learned in the BERT model is reduced, which can be regarded as miniaturization of the BERT model.

2.2 Learning method of BERT

The BERT model is primarily trained using MLM (Masked Language Model), but the model building time can be reduced by improving the training method. In PMI-Masking (Levine et al., 2021), a vocabulary set is set designed based on improved PMI (Pointwise Mutual Information) to facilitate the prediction of Mask words in MLM. We demonstrated how this learning strategy allows us to construct a high-performing BERT model with fewer epochs.

EarlyBERT (Chen et al., 2021) uses the Lottery Ticket Hypothesis (Frankle and Carbin, 2019) methodology used in computer vision tasks to identify structured winners in the early stages of BERT training result in a model with the same performance as conventional BERT ranging form 35% to 45% less cost.

The proposed method in this atudy learns BERT token embeddings from word2vec, with the token embeddings not having to be obtained by training the entire BERT model. The processing time for word2vec training is significantly less than that for BERT training.

2.3 Task specific BERT

TLM (Task-driven Language Modeling) (Yao et al., 2021) is a method used to learn BERT from scratch using only task-related data. The time required to build a model is 1/10-1/100 of that required for existing methods, and the performance of the built model outperforms that of general BERT models. Furthermore, the constructed models outperformed DAPT-TAPT (Gururangan et al., 2020), which is an excellent method for domain adaptation using BERT.

The method proposed in this study can be used task-specifically and directly in conjunction with TLM.

3 Method

In this study, we propose to fix the word embedding to reduce the construction time of BERT. The following procedures were used to

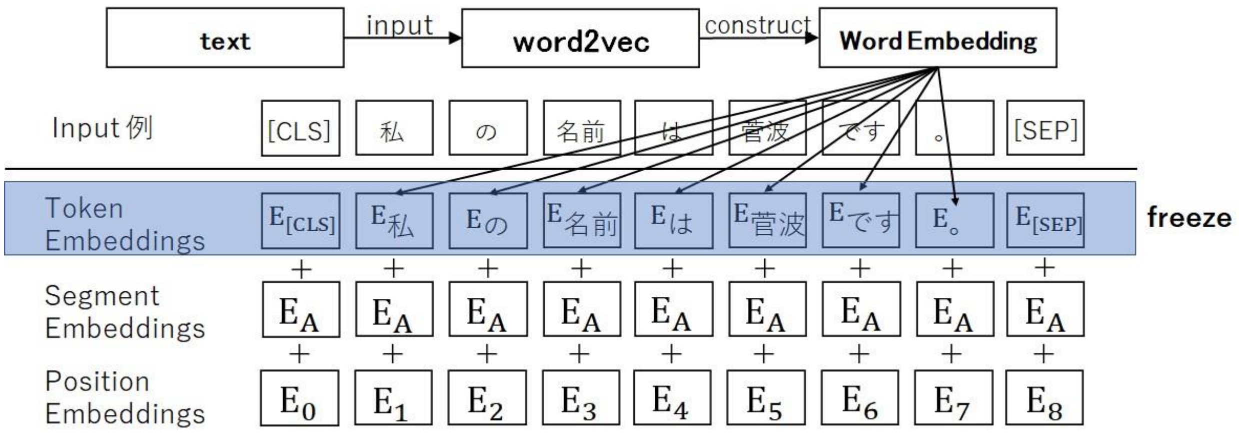
text
input
word2vec
construct
Word Embedding
Input 例
[CLS]
私
の
名前
は
菅波
です
。
[SEP]
Token Embeddings
$E_{[CLS]}$ $E_{私}$ $E_{の}$ $E_{名前}$ $E_{は}$ $E_{菅波}$ $E_{です}$ $E_{。}$ $E_{[SEP]}$
freeze
Segment Embeddings
E_A
Position Embeddings
E_0 E_1 E_2 E_3 E_4 E_5 E_6 E_7 E_8

Figure 1: Fixing Word Distributed Representations

achieve this. Figure 1 shows an image of the proposed method.

1. Construct word embeddings from the text using word2vec. This gives word embeddings for all words in the text.
2. Match the id of the word used for BERT training with the id of the word in the word embedding learned with word2vec.
3. When constructing BERT, the word embedding constructed in 1, is fixed as Token Embedding, and the parameters of Token Embedding are frozen so that they are not updated.

Word2vec fixes the parameters of Token Embedding to the learned word embedding by freezing them. The number of parameters can be reduced by 128 × 32,000 = 4,096,000 for a 128-dimensional 24-layer BERT and 1024 × 32,000 = 32,768,000 for a 1024-dimensional 4-layer BERT because the vocabulary size of the BERT is set to 32,000 in this study. The time required to learn BERT can be omitted, resulting in a shorter construction time.

4 Method

4.1 Construction of Word Embeddings

As the corpus for pre-training, we used dumped data from the Japanese Wikipedia as of November 1. These data are from Japanese Wikipedia, containing 1.17 billion characters with a data size of 3.26 GB. This data was text cleaned using a wikiextractor, the text was preprocessed, and multiple txt files were combined into a single text. The completed text was split into words using the "cl-tohoku/bert-base-japanese" tokenizer in Tohoku University's BERT. Then, we constructed a 128-dimensional word embedding and a 1024-dimensional word embedding from the text. using the Python library “gensim” word2vec. The learning algorithm used in this study was CBOW.

4.2 Construction of Japanese BERT

We built a 128-dimensional 24-layer Japanese BERT and a 1024-dimensional 4-layer Japanese BERT for 8 epochs each using a program for building Japanese BERT published on GitHub. Additionally, we modified the program to allow the construction of Japanese BERT by fixing the word variance representation constructed in section 4.1 to Token Embedding, and we used the proposed method to construct a 128-dimensional 24-layer Japanese BERT and a 1024-dimensional 4-layer Japanese BERT for 8 epochs each. The four Japanese BERTs constructed in this manner have a maximum input word count of 256 and a vocabulary of 32,000. The model building time was recorded for each epoch of model building. The flow of the Japanese BERT construction described so far is shown in Figure 2.

4.3 Model Evaluation

After constructing the Japanese BERT, we evaluated the model by comparing the percentage of correct answers in the document classification task. Livedoor news corpus published by Lonwit, Inc. was used for the evaluation data. The 7376

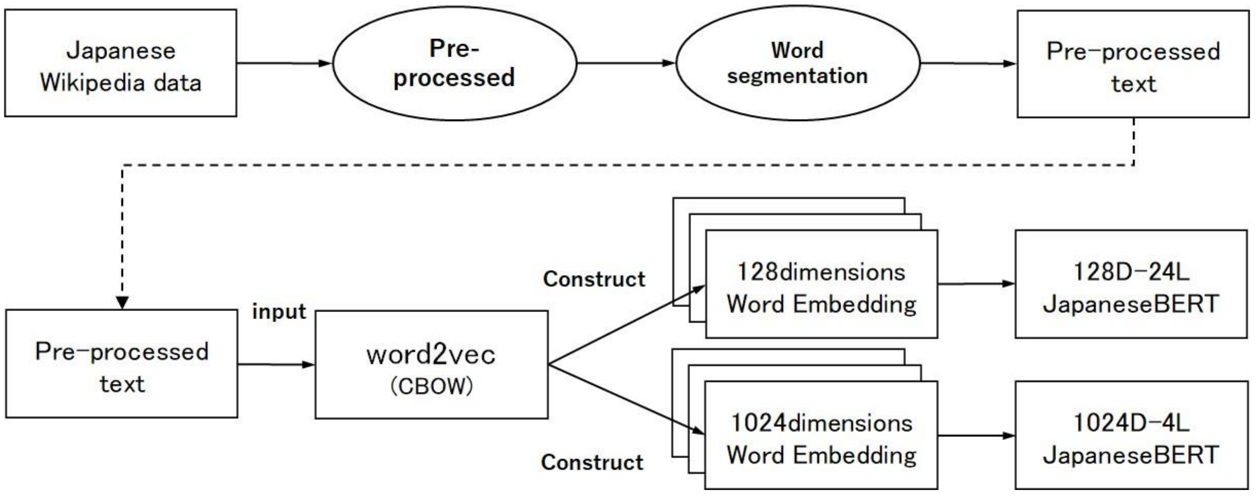

Figure 2: Flow chart for building Japanese BERT

articles in the corpus were assigned labels from 0 to 8 based on the category of the article, and the 7376 articles were assigned to the training data, validation data, and test data (Table 1).

label	category	train	valid	test
0	Dokujo tsushin	87	87	697
1	IT lifehack	87	87	697
2	Kaji Channel	86	86	693
3	Livedoor HOMME	51	51	410
4	MOVIE ENTER	87	87	697
5	Peachy	84	84	675
6	Smax	87	87	697
7	Sports Watch	90	90	721
8	Topic News	77	77	617
Total		736	736	5904

Table 1: Breakdown of data used

The model was trained using the training data, and the percentage of correct answers for the trained model was measured using the validation data after each training session. The loop of training and validation was set to a maximum of 15 epochs (Figure 3). When the highest percentage of correct answers to the validation data was not updated five times, the model training was terminated as Early Stopping, and the model with the highest percentage of correct answers up to that point was saved as the best model. Finally, the best model was subjected to document classification using test data to predict the category to which an article belongs based on the text of the article, and the percentage of correct answers was measured.

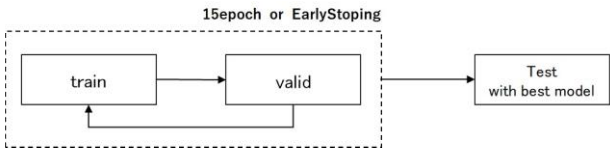

Figure 3: Model Evaluation

5 Result

5.1 Construction time

Table 2 shows the time per epoch taken to build the Japanese BERT. The "H" in the model name represents the dimensionality of the word embedding, and the "L" represents the number of layers. "N" indicates that the construction was performed using the conventional method, whereas "W" indicates that the construction was performed using the proposed method. The proposed method reduces construction time by 4 min per epoch for 24 layers of 128 dimensions and by 27 min per epoch for 4 layers of 1024 dimensions.

Model	Construction Time
H128-L24-N	16h49m
H128-L24-W	16h45m
H1024-L4-N	18h05m
H1024-L4-W	17h38m

Table 2: Construction time per epoch

5.2 Document Classification Accuracy

Table 3 shows the percentage of correct answers in the document classification task performed using the constructed Japanese BERT. The best correct response rate is indicated by bolded numbers. The proposed method did not improve the accuracy of the 24 layers with 128 dimensions, but it improved the accuracy of the 4 layers with 1024 dimensions.

6 Result

6.1 Model building Time

Table 2 shows that the proposed method reduces the construction time by 4 min for 24 layers of 128 dimensions and by 27 min for 4 layers of 1024 dimensions, indicating a reduction in construction time due to the omission of learning word embeddings in BERT. The fact that the time reduction was greater in the 1024-dimensional 4-layer case indicates that the larger the dimensionality of the Japanese BERT, the greater the number of parameters that can be reduced, thus the greater the time reduction by the proposed method.

6.2 Model Accuracy

Table 3 shows that the proposed method improved the accuracy of the 1024-dimensional, 4-layer model. The best scores were also recorded early by the proposed method for both BERTs. Therefore, we assumed that the accuracy of Japanese BERT fixed with the word embedding created using word2vec will converge at earlier epochs. If accuracy can be achieved at an early stage while improving accuracy in this way, it is possible to build a highly accurate model in fewer epochs when constructing BERT compared to the conventional method, therefore, the construction time may be significantly reduced.

7 Conclusion

To reduce the construction time of BERT, this study focuses on the time required to learn word embedding during BERT construction, and proposes a method to construct Japanese BERT by fixing a previously learned word embedding using word2vec to Token Embedding. In the experiment, we constructed a 128-dimensional 24-layer and a 1024-dimensional 4-layer Japanese BERT and measured the construction time, and found that the construction time could be reduced. The accuracy measurements in a document classification task showed that accuracy improved and converged at a lower number of epochs. This shows that fixing the word embedding can reduce the construction time of BERT.

	H128 L24-N	H128 L24-W	H1024 L4-N	H1024 L4-W
ep1	0.7033	0.6982	0.6386	**0.7236**
ep2	0.7100	0.7060	0.6811	0.7107
ep3	0.6922	**0.7148**	0.6765	0.6816
ep4	0.6865	0.6738	0.6822	0.6685
ep5	0.7122	0.6963	0.6811	0.6992
ep6	0.6729	0.6975	0.7041	0.7002
ep7	0.7029	0.6897	**0.7043**	0.6753
ep8	**0.7204**	0.7126	0.6941	0.6692

Table 3: Document Classification Accuracy

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186.

Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. International Conference on Learning Representations.

Hinton, Geoffrey and Vinyals, Oriol and Dean, Jeff. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Yoav Levine and Barak Lenz and Opher Lieber and Omri Abend and Kevin Leyton-Brown and Moshe Tennenholtz and Yoav Shoham. 2021. {PMI}-Masking: Principled masking of correlated spans. International Conference on Learning Representations.

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. ICLR 2013.

Zafrir, Ofir and Boudoukh, Guy and Izsak, Peter and Wasserblat, Moshe. 2019. Q8bert: Quantized 8bit bert. arXiv preprint arXiv:1910.06188.

Hassan Sajjad and Fahim Dalvi and Nadir Durrani and Preslav Nakov. 2020. Poor Man’s BERT: Smaller and Faster Transformer Models. CoRR, abs/2004.03844.

Or Sharir, Barak Peleg, Yoav Shoham. 2020. The Cost of Training NLP Models : A Concise Overview CoRR abs/2004.08900.

Gururangan, Suchin and Marasovic, Ana and Swayamdipta, Swabha and Lo, Kyle and Beltagy, Iz and Downey, Doug and Smith, Noah A. 2020. Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8342–8360, Association for Computational Linguistics.

Sanh, Victor and Debut, Lysandre and Chaumond, Julien and Wolf, Thomas. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Chen, Xiaohan and Cheng, Yu and Wang, Shuohang and Gan, Zhe and Wang, Zhangyang and Liu, Jingjing. 2021. EarlyBERT: Efficient BERT Training via Earlybird Lottery Tickets. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2195–2207, Association for Computational Linguistics.

Jiao, Xiaoqi and Yin, Yichun and Shang, Lifeng and Jiang, Xin and Chen, Xiao and Li, Linlin and Wang, Fang and Liu, Qun. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. Association for Computational Linguistics, pages 4163–4174.

Yao, Xingcheng and Zheng, Yanan and Yang, Xiaocong and Yang, Zhilin. 2021. NLP From Scratch Without Large-Scale Pretraining: A Simple and Efficient Framework. International Conference on Learning Representations.

Lan, Zhenzhong and Chen, Mingda and Goodman, Sebastian and Gimpel, Kevin and Sharma, Piyush and Soricut, Radu. 2019. Albert: A lite bert for selfsupervised learning of language representations. arXiv preprint arXiv:1909.11942.

Sun, Zhiqing and Yu, Hongkun and Song, Xiaodan and Liu, Renjie and Yang, Yiming and Zhou, Denny. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2158–2170, Association for Computational Linguistics.