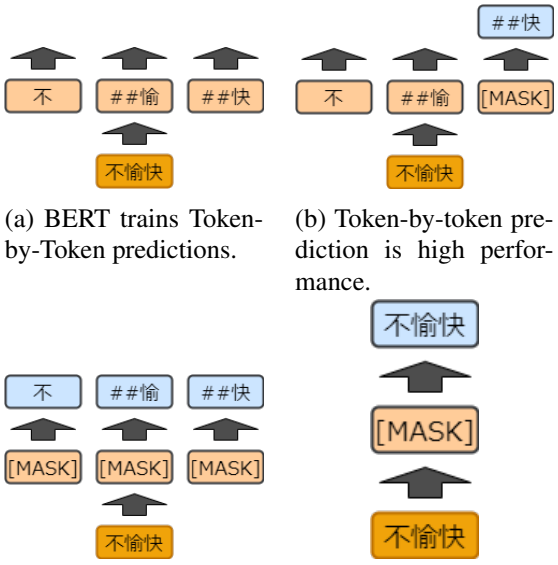
Vocabulary expansion of compound words for domain adaptation of BERT

Hirotaka Tanaka **Hiroyuki Shinnou**
Department of Computer and Information Sciences Ibaraki University
Hitachi, JAPAN
{ 22nd304a, hiroyuki.shinnou.0828 } @vc.ibaraki.ac.jp

Abstract

Pretraining models such as BERT, have achieved high accuracy in various natural language processing tasks by pretraining on a large corpus and fine-tuning on downstream task data. However, BERT trains token-level inferences, which make it difficult to train unknown or compound words that are split by byte-pair encoding. In this paper, we propose an effective method for constructing word representations in vocabulary expansions for such compound words. The proposed method assumes domain adaptation by additional pretraining and expands the vocabulary by embedding a synonym as an approximate embedding of additional words. We conducted experiments using each vocabulary expansion method and evaluated these experiments for their accuracies in predicting additional vocabularies in the masked language model.



(a) BERT trains Token-by-Token predictions.

(b) Token-by-token prediction is high performance.

(c) Difficult to predict the entire splitted representation.

(d) The entire representation as a single Token is easily predicted.

Figure 1: Examples of vocabulary expansion targets

1 Introduction

Pre-learning models have significantly improved the performances of various natural language processing systems (Peters et al., 2018)(Radford et al., 2018). Bidirectional encoder representations from transformers (BERT)(Devlin et al., 2019) is pretrained model that consists of a stacked multi-head attention in the Transformer(Vaswani et al., 2017). A BERT outputs word representations that embed the context of the input word sequence.

Pre-learning models have domain adaptation problems, and they perform various downstream tasks with high accuracy by applying models pretrained on a large corpus to the downstream task data. Therefore, if the domain of the downstream task differs significantly from the domain of the pretrained corpus, the solution of the downstream task will have low accuracy. Gururangan et al.(2020) proposed a method for domain adaptation by additional pretraining on a corpus of downstream task domains.

Domain adaptation problems also appear in the vocabulary. The vocabulary covered by the pretrained model depends on the pretrained corpus. Therefore, it must adapt to the vocabulary appearing in the downstream task data through vocabulary expansion. The adapt-and-distill approach by

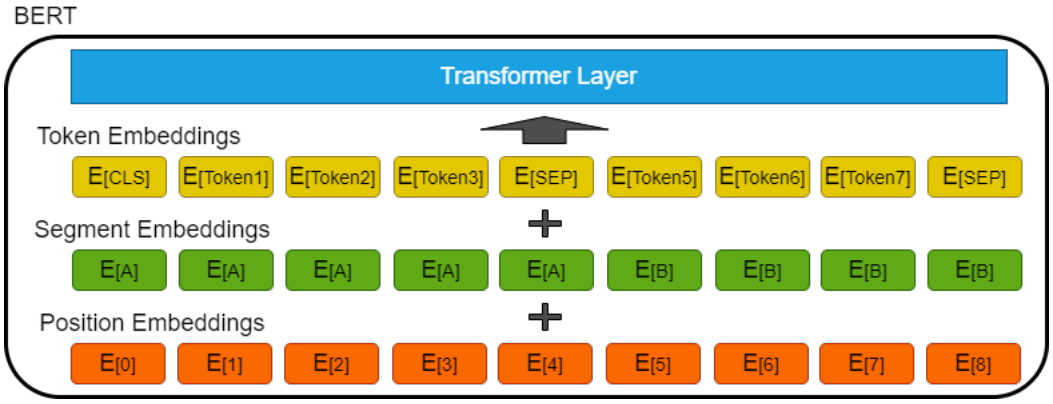


Figure 2: Construction of BERT input embeddings

Yao et al.(2021) and AVocaDo strategy by Hong et al.(2021) are vocabulary expansion methods for domain adaptation in the pretraining models.

BERT trains token-level expressions, and it has difficulty training the expressions of compound words, named entities, and phrases. The Japanese vocabulary for a standard BERT model is token units, and tokenization is performed by morphological analysis and byte-pair encoding. The division of unknown words into known words makes it possible to cover a large number of words using a small vocabulary. However, the masked language model (MLM) in the pretraining task trains for token by token prediction. Therefore, it is difficult to predict an entire representation constructed using multiple tokens (see Figure 1).

Yanada et al.(2020) proposed the pretraining model LUKE for training the representations of entities constructed from multiple words. LUKE provides entity embeddings in addition to ordinary word embeddings and models and trains the relationship between ordinary tokens and entities by the entity-aware self-attention mechanism. However, LUKE requires expensive pretraining.

In this paper, we propose a method to add a vocabulary to the BERT model. In vocabulary expansion, the focus is on the method for constructing word embeddings in the additional vocabulary. By assuming domain adaptation through the additional pretraining of downstream task data, we expect the model to train word embeddings of the additional vocabulary based on approximate vectors.

2 Related work

2.1 Token embeddings for BERT

Tokens obtained from the input sentences are converted to token embeddings. The BERT input vector consists of three embeddings (see Figure 2). Token embeddings represent words. Segment embeddings embed information that identifies each sentence from among the multiple input sentences, and position embeddings represent the token's position in the input.

The final output of BERT is a contextualized word representation. However, the embedding used as input to BERT is unique for each token.

2.2 MLM

MLM is a pretraining method of BERT. The task was to predict the tokens replaced by MASK tokens. The standard approach is to replace 15% of the input tokens. These 15% input tokens include the following replacements:

- 80% are replaced with the special token `[MASK]`.
- 10% are replaced by other random tokens.
- The remaining 10% tokens are kept intact.

3 Vocabulary expansion for BERT

We expand the vocabulary of the pretrained BERT model by adding word embeddings to token embeddings. Therefore, the challenge for the vocabulary expansion methods is to obtain additional word embeddings corresponding to the BERT embedding space.

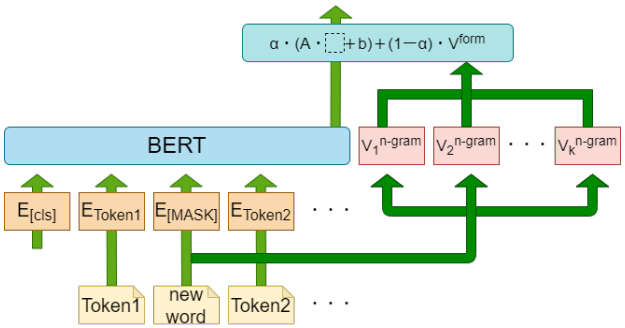


Figure 3: SHALLOW method of BERTRAM

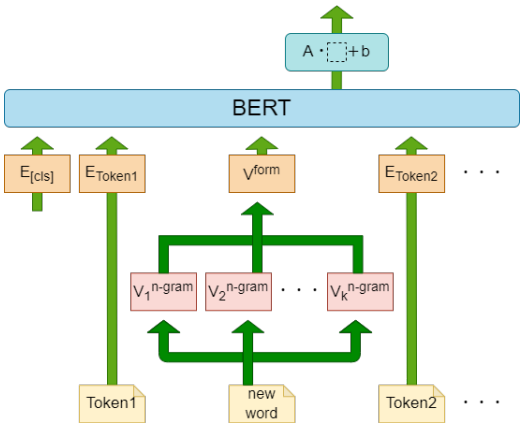


Figure 4: REPLACE method of BERTRAM

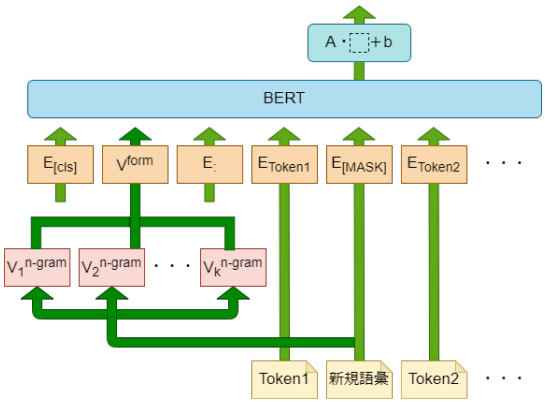


Figure 5: ADD method of BERTRAM

3.1 Static word representations

A method for obtaining additional word embeddings is to use static word representations from distributed representation models such as Word2Vec or fastText(Bojanowski et al., 2017)(Joulin et al., 2016). When a distributed representation model trained on the target word exists, vocabulary expansion is achieved by adding the model to the BERT embeddings. Even when a distributed representation model with trained target words is absent, the calculation cost is lower to train a distributed representation model that includes the target words than to pretrain a BERT model from scratch.

In particular, we first prepare a distributed representation model trained on the additional vocabulary. Next, the transformation model trains a mapping from the distributed representation model to the BERT word embedding based on the vocabulary set commonly trained by them. Mikolov et al.(2013) proposed a method for training the mapping by applying stochastic gradient descent to reduce the mean squared error between the source and target word vectors.

3.2 Mean vector of subwords

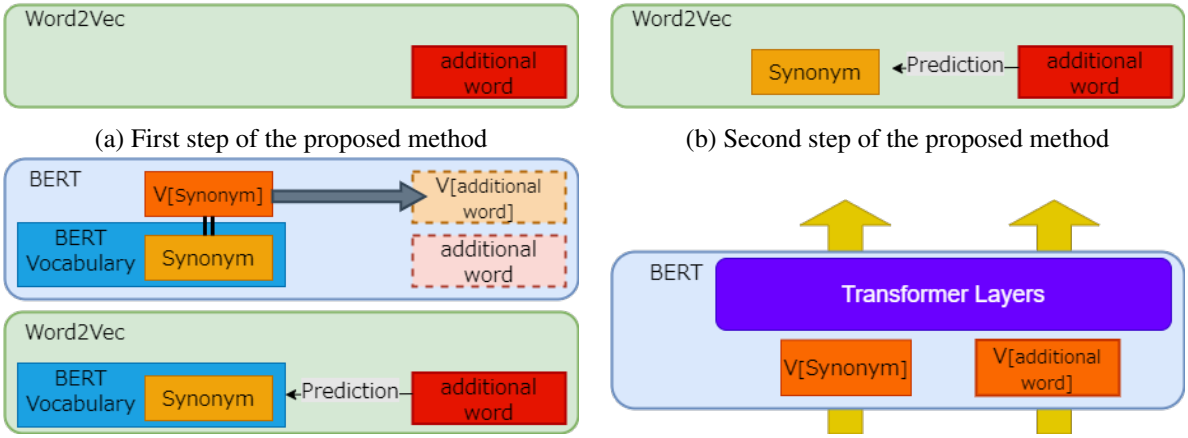
The mean vectors of subwords are used to obtain embeddings using only pretrained BERT models. This method is also used in adapt-and-distill approach(Yao et al., 2021).

The target words of the expansion are processed with multiple tokens in BERT. The mean vector of these token embeddings is calculated from the pretrained BERT model. Let the mean vector be the additional word embeddings.

3.3 BERTRAM

Schick and Schütze (2020) proposed BERT for attentive mimicking (BERTRAM) as a method for obtaining additional word embeddings from the output of the pretrained BERT. BERTRAM focuses on adding low-frequency words in the pretrained corpora and trains additional word embeddings in a form-context model. The form model trains on a character basis, whereas the context model trains on a context basis. The form model trains character n-gram embeddings and constructs additional word embeddings from them. The context model trains additional word embeddings by predicting masked additional words in a sentence, which is similar to the method used by the standard BERT model to train contextualized word embeddings.

For the context model construction, Schick et al. tried three methods: SHALLOW, REPLACE, and ADD.



(a) First step of the proposed method

(b) Second step of the proposed method

(c) Third step of the proposed method

(d) Final step of the proposed method

Figure 6: Schematic figure of the proposed method

In the SHALLOW method, the output of the pre-trained BERT for the mask token is obtained as the contextualized word embedding of the target word (see Figure 3).

In the REPLACE method, the embedding by the form model is input to the pre-trained BERT as a target word embedding. The output of the BERT is then obtained as a contextualized word embedding representation of the target word (see Figure 4).

The ADD method is a combination of the SHALLOW and REPLACE methods. First, the target words are replaced with mask tokens. Then, a word embedding using the form model and the word ":" are added at the beginning of the sentence. The resulting sentence is input to the pre-trained BERT, and the embedding corresponding to the mask token in the output is obtained as a contextualized word embedding of the target word (see Figure 5).

In the case that the context model method is SHALLOW, the embeddings of the form-context model are computed as:

$$v_{(w,C)} = \alpha \cdot (A \cdot v^{context}_{(w,C)} + b) + (1-\alpha) \cdot v^{form}_{(w,C)} \quad (1)$$

where w is a target word, C is a sentence set, and $A \in \mathbb{R}^{d \times d}$ and $b \in \mathbb{R}^{d}$ are trainable parameters.

In the case that the context model method is REPLACE or ADD, the embedding of the form model output is an input to the context model. Therefore, the embedding of the form-context model is obtained by linear transformation of the context model output:

$$v_{(w,C)} = (A \cdot v^{context}_{(w,C)} + b). \quad (2)$$

The loss on training is computed as follows:

$$\|e_w - v_{(w,C)}\| \quad (3)$$

where e_w is the BERT token embedding corresponding to word w.

BERTRAM training is a three-step process. When the context model method is ADD, the training method involves the following steps:

1. Train only the context model using the SHALLOW method.
2. Train only the form model.
3. Train the entire parameters of the BERTRAM model constructed by the context model using the ADD method.

When training BERTRAM with the ADD and REPLACE methods, the SHALLOW method is also used to trian the context model in the first step.

The training parameters of the pretrained BERT model are frozen during all training steps.

In the BERTRAM experiment, the additional vocabulary "____" is added as a special token denoted by "<BERTRAM:____>." When using word embeddings added by BERTRAM, the additional vocabulary in the data is replaced with "<BERTRAM:____>" representations.

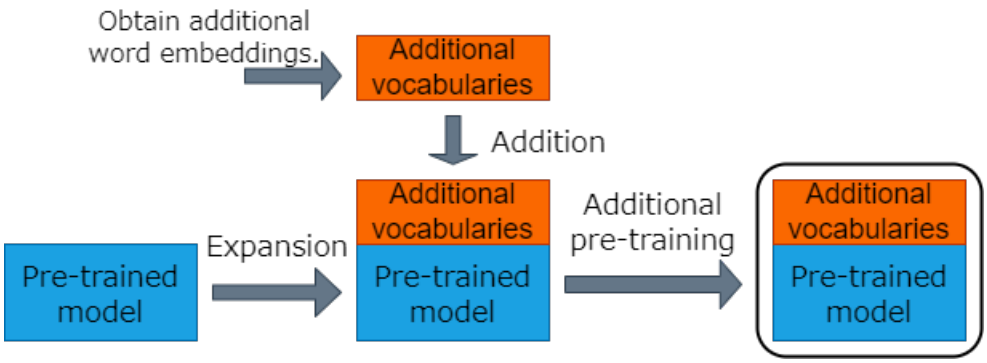


Figure 7: Vocabulary Expansion Process

4 Proposed method

Domain adaptation to the downstream task domain enhances the performance of the pre-learning model for downstream tasks. We expect that when adapting the model to the domain using the method of Gururangan et al.(2020) even though the additional word embeddings are approximate, the additional pretraining can learn the appropriate embeddings.

Approximate embeddings of additional words can be obtained from their synonyms. Additionally, the synonyms included in the pretrained BERT vocabulary have their embedded representations already learned by BERT. Therefore, in the proposed method, we add the synonym embeddings included in the pretrained BERT vocabulary to the model as the additional word embeddings. Then, the model additionally pretrains on the downstream task data.

We applied distributed representation models such as Word2Vec, for synonym estimation. The distributed representation model is a model trained on both the additional vocabulary and BERT vocabulary.

At the vocabulary expansion stage, the synonym embeddings and additional word embeddings are identical. By additionally pretraining the entire training parameters, including BERT's token embeddings, the model trains the optimal embedding for each model. Additional pretraining uses the downstream task domain as the training data, which fine-tuning the entire training parameters of the model with the MLM.

In the proposed method (see Figure 6), the model fine-tunes the word embeddings with MLMs based on the synonym embeddings of the additional vocabulary.

The method involves the following steps:

1. A distributed representation model such as Word2Vec, is trained with the additional vocabulary.
2. The distributed representation model estimates the synonyms for the additional vocabulary.
3. The token embeddings of the estimated synonyms in the BERT vocabulary are included as additional word embeddings.
4. The BERT with the expanded vocabulary is additionally pretrained with the MLM on a corpus containing the additional vocabulary.

5 Experiments

5.1 Methodology

In this experiment, we extended the pretrained Japanese BERT vocabulary(see Figure 7). In addition to the proposed method, we compared the method using static word embedding, the method of mean vectors of subwords, and BERTRAM.

The proposed method applies cosine similarity to the similarity between additional vocabularies and synonyms.

The experiments for all methods were conducted under the following conditions:

- MeCab was applied for the morphological analysis.
- The models added to the vocabulary by each method were additionally pretrained on the downstream task domain data by the MLM.

Target words	Synonyms	Similarities
殺人事件	殺人	0.8607
社会科学	人文	0.8482
推理小説	ミステリ	0.8147
自分自身	自分	0.8105
彼女自身	彼女	0.8102
日本文学	国文学	0.8080
新聞記者	記者	0.8036
長編小説	小説	0.8006
短編小説	小説	0.7950
携帯電話	携帯	0.7866
近親相姦	レズ	0.7858
少年時代	幼少	0.7831
金融機関	銀行	0.7758
統合失調症	うつ病	0.7750
精神医学	臨床	0.7687
日常生活	日常	0.7675
成果主義	デフレ	0.5805
地球温暖化	エコ	0.5796
練習問題	文法	0.5764
人生経験	悩み	0.5706
自己満足	信念	0.5691
宣伝文句	キャッチフレーズ	0.5649
参考文献	文献	0.5636
古今東西	落語	0.5500
携帯小説	妄想	0.5477
行政書士	弁護士	0.5416
設定資料集	プラモデル	0.5342
成功法則	生き方	0.5338
日本語版	翻訳	0.5316
自己責任	考え方	0.5269
不完全燃焼	不発	0.5246
裁判員制度	陪審	0.5194

Table 1: Examples of synonyms and similarities

- The evaluation method estimated the new vocabulary by replacing only the new vocabulary with the MASK tokens.
- The evaluation index was the mean reciprocal rank (MRR).
- The evaluation value was the average of five trials with different random seeds.

5.2 Japanese pretrained BERT model

We used the model named cl-tohoku/bert-base-japanese published by the Inui Laboratory at Tohoku University as the pretrained BERT. This model is available in Hugging Face's transformers library. The model was pretrained using the Japanese Wikipedia as the pretraining corpus.

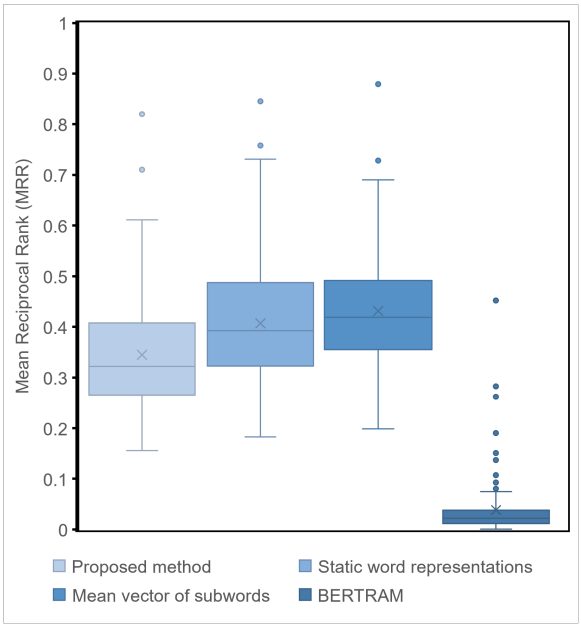


Figure 8: The experimental results

5.3 Distributed representations

For static word embeddings, we used the Japanese Wikipedia entity vectors available at `http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/`. This vector was trained using the Word2Vec model.

5.4 BERTRAM

To learn BERTRAM, we applied the programs adapted to the Japanese language by referring to the author-implemented programs of the BERTRAM paper available at the following two websites: `https://github.com/timoschick/bertram` and `https://github.com/timoschick/form-context-model` . This BERTRAM model was trained on the Japanese Wikipedia.

5.5 Datasets

The dataset for this experiment is sourced from the Amazon Review Corpus. This corpus is available at `https://webis.de/data/webis-cls-10.html`.

This corpus was divided into three domains according to the type of product: Books, DVDs and

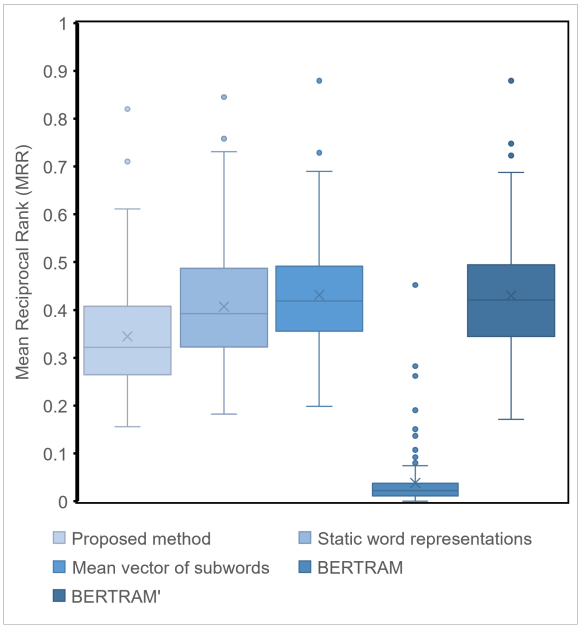


Figure 9: BERTRAM tokenization experiment

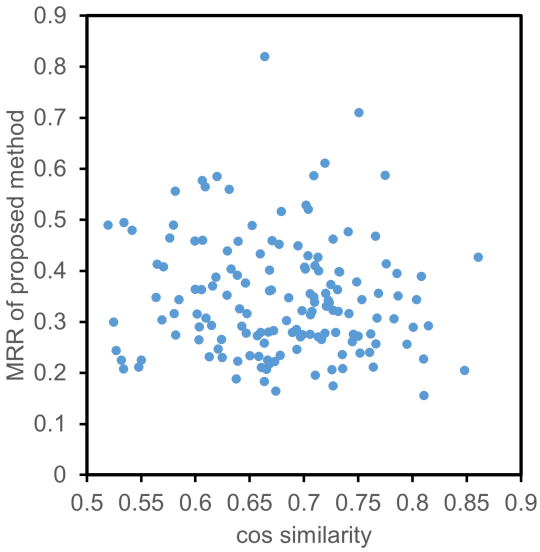


Figure 10: Similarity and accuracy for each word

Music. In this experiment, we used the Books domain. The corpus included unlabeled reviews and reviews labeled with stars. We used unlabeled data from the Books domain.

We defined the additional target words as having four or more Kanji characters, which were defined as general proper nouns in the Mecab Ipadic NEologd dictionary. There were 149 such words in the Amazon review corpus. These words were divided into multiple tokens in the pretrained BERT and into multiple morphemes in the standard MeCab. The dataset had 100 sentences of the corpus for each word. For each word, 50 sentences were those of the training data and the other 50 were the test data. The training and test data had 7,450 sentences each.

6 Results and discussion

6.1 Experimental results

Examples of synonyms and similarities obtained using the proposed method are shown in Table 1, and this table presents the top 16 and bottom 16 similarities.

The experimental results for each method are shown in Figure 8. This figure presents the MRR in each word as a box-and-whisker plot. Although the proposed method had relatively low accuracy, its prediction accuracy was as good as that of other methods. There was no significant difference between the static word representations and mean vector of the subwords. However, BERTRAM's prediction accuracy was significantly low.

6.2 Tokenization of BERTRAM

We identify the causes of BERTRAM's experimental results. Tokenization in BERTRAM is different from other methods used for additional vocabulary. BERTRAM initially picks out the additional vocabulary in the sentence as special tokens and splits the sentence before and after it. Then, it tokenizes each of the split sentences.

This tokenization works in English, but it does not work in Japanese. Japanese requires morphological analysis. When BERTRAM's tokenization is applied to Japanese, it morphologically analyzes each segmented sentence. The morphological analyzer does not receive a complete sentence, which results in low analytical accuracy.

We conducted an experiment to confirm this. We let BERTRAM' be the method of morphological analysis of complete sentences similar to other methods. The results of this experiment are shown in Figure 9. BERTRAM' was as accurate as the static word representations and mean vectors of subwords. Therefore, the low accuracy of BERTRAM resulted from tokenization.

6.3 Relationship between similarity and prediction accuracy

To validate the effectiveness of the proposed method for each word, we analyzed the relationship between the similarities and accuracies of the synonyms. A graph that plots the similarities and accuracies for the words is shown in Figure 10. The correlation coefficient between the similarities and accuracies is -0.0815. Therefore, there was no significant correlation between the similarities and accuracies if synonyms.

Using the accuracy of each method for each word, it was not possible to determine the most effective method for all words. Futher analysis is required to determine the specific word features that would be effective for each method.

7 Conclusion

In this paper, we examined how to apply an effective vocabulary expansion method to compound words. We proposed a method using synonyms by assuming domain adaptation with additional pretraining. An analysis of the relationship between the prediction accuracy and similarity of the synonyms to the target word revealed no significant correlation between them. Furthermore, we improved the tokenizer implemented in BERTRAM to apply it to the Japanese language. Our future work will include an analysis of the features of compound words, and we will propose effective methods for compound words. We also propose to extend the method to sequence representations longer than those used in this study.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July. Association for Computational Linguistics.

Jimin Hong, TaeHee Kim, Hyesu Lim, and Jaegul Choo. 2021. AVocaDo: Strategy for adapting vocabulary to downstream domain. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4692–4700, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759*.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Technical report, OpenAI.*

Timo Schick and Hinrich Schütze. 2020. BERTRAM: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Online, July. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online, November. Association for Computational Linguistics.

Yunzhi Yao, Shaohan Huang, Wenhui Wang, Li Dong, and Furu Wei. 2021. Adapt-and-distill: Developing small, fast and effective pretrained language models for domains. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 460–470, Online, August. Association for Computational Linguistics.