Evaluation of Pretrained BERT Model by Using Sentence Clustering

Naoki Shibayama Rui Cao Jing Bai Wen Ma Hiroyuki Shinnou
Ibaraki University, Department of Computer and Information Sciences
4-12-1 Nakanarusawa, Hitachi, Ibaraki JAPAN 316-8511
{19nm714t, 18nd305g, 19nd301r, 19nd302h, hiroyuki.shinnou.0828}
@vc.ibaraki.ac.jp

Abstract

For evaluation of pre-trained models like bidirectional encoder representations from transformers (BERT), task-based approaches are frequently adopted and there is a possibility that meta parameters for fine-tuning influence results of the evaluations. However, task-based approaches for languages, except English, have a problem- there is no common dataset for their evaluation. Hence, evaluating pre-trained models for these languages with task-based approaches is challenging. In this work, we evaluate Japanese pre-trained BERT models with CLS token. We input labeled sentences to models, get CLS token embeddings, and calculate scores from in-class and out-of-class dispersions, which can be calculated from embeddings and labels of sentences. Experiment results show that a model released by Laboro.AI Inc. is the best Japanese pre-trained BERT model. Meanwhile, the results of evaluation with sentence clustering are different from those of evaluations that are based on fill mask task.

1 Introduction

BERT (Devlin et al., 2019) is a high-performance pre-training model. It helped in the improvement of the performance of natural language processing tasks. Generally, task-based approaches were adopted for evaluating pre-training models like BERT. In English language, a dataset for task-based evaluation, such as the general language understanding evaluation (GLUE) (Wang et al., 2018), can be used, and it is easy to compare models. However, when a pre-trained model is fine-tuned for task-based evaluation, meta parameters for fine-tuning may influence scores of the model. Hence, task-based evaluation with fine-tuning has a possibility of biased evaluation. Also, there is no common task-based dataset for languages except English, so it is challenging to compare pre-trained models for other languages.

In this work, we evaluate Japanese pre-trained BERT models using CLS token embeddings in outputs of target models. CLS token embedding can be regarded as an input sentence embedding, and models can be rated with evaluating embeddings itself. However, how to evaluate sentence embeddings is also challenging. Here, we use clustering to evaluate sentence embeddings. Also, we prepare sets of sentences sorted by genre and use BERT models to get embeddings of each sentence. Then, we cluster those embeddings and evaluate models with clustering score.

2 Related Works

Generally, a task-based approach for evaluation is adopted to compare and evaluate pre-trained models like BERT. Although this simple method requires data for evaluation, it consists of the following 3 steps:

1. Solve a task with pre-trained model A and get its accuracy.

2. Solve this same task with pre-trained model B and get its accuracy.

3. Compare the accuracies and evaluate models A

and B.

The GLUE can be used for English, but there is no common dataset for other languages, so we have to prepare the dataset for evaluation ourselves.

There is a work that compared and evaluated some Japanese pre-trained BERT models. In this work, we evaluated three BERT models using document classification tasks with the Amazon dataset (Shibayama et al., 2019). However, BERT is a model for sentences, and there is no established method of document classification with BERT. Therefore, whether document classification is the right task to evaluate or not is questionable. We use a sentence as input of BERT and evaluate models using CLS token embeddings, which can be considered as sentence embeddings from outputs of BERT.

The approaches for evaluation of embeddings are task-based, but in the case of word embeddings from outputs of some method like word2vec (Mikolov et al., 2013), there is a viewpoint that embeddings represent the meaning of words. Also, there is a research that evaluated embeddings with correlation of similarities between words calculated from the similarity of embeddings and by hand (Sakaizawa and Komachi, 2016).

3 Evaluation of BERT

In Section 2, we mentioned that a task-based approach is frequently adopted to evaluate embeddings. Also, we mentioned that there is a viewpoint that embeddings represent the meaning of words. When this viewpoint is applied to clustering, we can say that a cluster can be represented by a group of embeddings in it. In what follows, we use this to evaluate pre-trained BERT models with sentence clustering.

3.1 Method of the Evaluation

Embeddings that were outputted from BERT model m, were evaluated by the following 5 steps. Labels for sentences of model m's input were required to do this evaluation.

1. Get CLS token's embedding from the output of each sentence of model m, and use the embedding as the sentence vector.
2. Check which class contains the sentence vector, and calculate $g_i^{(m)}$: centroid of each class of model m.
3. Calculate A_m: in-class dispersion of each class from the following expression[1].

$$A_m = \sum_{i=1}^{N} \sigma_i^2 \tag{1}$$

   where $\sigma_i^2 = \sum_{j \in C_i} ||g_i^{(m)} - x_{i,j}||^2$, C_i is class i and N is number of classes.

4. Calculate $g^{(m)}$: average centroids of all classes and calculate B_m: out-of-class dispersion from the following expression.[2]

$$B_m = \sum_{i=1}^{N} ||g^{(m)} - g_i^{(m)}||^2 (N = \text{Number of classes}) \tag{2}$$

5. Calculate a degree of separation: $M_m = \frac{A_m}{B_m}$, and use M_m as a score of model m. This score becomes smaller when clustering with model m is performed properly.

Figure 1 summarizes the flow of the evaluation.

3.2 Re-evaluation by Using Fill Mask Task

We re-evaluated models with a fill mask task in order to verify the results of sentence clustering evaluation. The steps for the re-evaluation are as following:

1. Prepare a dataset- we prepared a dataset that contains sentences and which word to be masked in matching sentence as labels.
2. Predict masked word with model- we calculated percentages that mask token was the word in matching label which was defined in a dataset from outputs of models.
3. Average and comparison- we compared averages of percentages that were calculated in step 2.

[1] We consider the second power of deviation as the dispersion in this work in order to calculate easily. So, true in-class dispersion can be calculated from σ_i^2/N.

[2] Also, we consider the second power of deviation of centroids of all classes as dispersion like A_m.

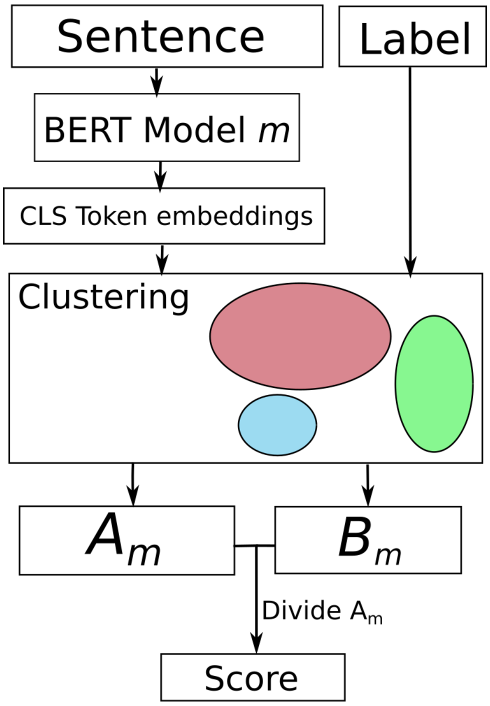

Figure 1: The flow of evaluation with sentence clustering

Detailed information on the abovementioned comparison is in the next subsection.

3.3 Experimental Setups

Firstly, we explain common setups for evaluation with sentence clustering and fill mask task. We compared six models: a model which was released by Kurohasi & Murawaki Lab at Kyoto University (hereafter, Kyoto Univ. Ver.)[3], Mr. Morinaga (hereafter, MeCab Ver.)[4], Mr. Yohei Kikuta (hereafter, SP Ver.)[5], Inui & Suzuki Lab at Tohoku University (hereafter, Tohoku Univ. Ver.)[6], National Institute of Information and Communications and Technology (NICT) (hereafter, NICT Ver.)[7], and Laboro.AI Inc.(hereafter, Laboro Ver.)[8]. We did not fine-tune models for comparisons.

Table 1 summarizes the word tokenizer and pre-training corpus of pre-trained models. Model size of all models are base format of official BERT(Devlin et al., 2019): 12-layer, 768-hidden, and 12-heads. So sentence vectors we got in the evaluation with sentence clustering have 768 dimentions. Juman++ and MeCab are software for morphological analysis. Juman++ uses Recurrent Neural Network Language Model and MeCab uses bi-gram Markov model for analysing. SentencePiece is unsupervised text tokenizer and detokenizer, so model publishers which use SentencePiece as tokenizer release pre-trained SentencePiece model with their BERT model.

Table 1: Information of pre-trained BERT models

Models	Tokenizer (characteristic)	Pre-training corpus
Kyoto Univ. Ver.	Juman++	Wikipedia
MeCab Ver.	MeCab + NEologd (No subword tokenize)	Articles of business news
SP Ver.	SentencePiece (do_lower_case = True)	Wikipedia
Tohoku Univ. Ver.	MeCab + NEologd	Wikipedia
NICT Ver.	MeCab + Jumandic	Wikipedia
Laboro Ver.	SentencePiece	Texts on the Internet (12GB)

In the evaluation with sentence clustering, we used Livedoor news corpus[9]. This dataset contains nine categories of articles and we used one hundred articles per category. We extracted titles from selected articles and regarded categories as classes. Then, we calculated scores with the method in Section 3.1 and compared these scores.

In the evaluation with fill mask task, we made a fill mask dataset from Japanese domain of Webis-

[3] http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT日本語Pretrainedモデル We used normal version.

[4] https://qiita.com/mkt3/items/3c1278339ff1bcc0187f

[5] https://github.com/yoheikikuta/bert-japanese

[6] https://github.com/cl-tohoku/bert-japanese This model can be used easily as "cl-tohoku/bert-base-japanese" from HuggingFace's transformers (Wolf et al., 2019), and we used it.

[7] https://alaginrc.nict.go.jp/nict-bert/index.html We used the byte pair encoding (BPE) version.

[8] https://laboro.ai/column/laboro-bert/

[9] http://www.rondhuit.com/download.html#ldcc

CLS-10 (Prettenhofer and Stein, 2010) and used it. The following two steps show how to make a fill mask dataset from Webis.

1. Pick twenty nouns that have the highest frequencies of occurence from the test data of each domain: books, DVDs, and music.
2. Pick five sentences that contain matching selected words from test data of the matching domain randomly to each selected word.
3. Use nouns which were selected in step 1 as labels for matching sentences which were selected in step 2.

We replaced “selected nouns” which appeared for the first time in matching sentence with mask token. The following shows selected Japanese nouns for each domain.

books: 本, 人, 著者, 内容, 自分, 作品, 本書, 感じ, 文章, 主人公, 小説, 部分, 最後, 言葉, 読者, 作者, 人間, 物語, 他, 世界
DVDs: 映画, 作品, 人, シーン, 映像, 原作, 自分, ストーリー, 内容, ファン, 感じ, 主人公, 最後, アニメ, ドラマ, 物語, 人間, 世界, 子供, 部分
music: 曲, アルバム, 作品, 人, 音楽, 感じ, ファン, 音, バンド, 自分, 歌詞, 声, ギター, 歌, ＣＤ, 楽曲, サウンド, ライブ, シングル, 前作

We calculated percentages that mask token is the word in matching label with prepared dataset and each model. Then, we averaged percentages and compared these. The following shows notices of this comparison.

- We used transformers (Wolf et al., 2019) to solve the fill mask task.
- We replaced “ＣＤ” (Fullwidth form of CD) with CD (Halfwidth form of CD).
- We lowercased sentences when we used SP Ver. model. We did not lowercase sentences when we used SP Ver. model for the first time, and the model tokenized CD as token “C” and token “D”. We checked vocabulary file of the model and found the word “cd”, not “CD”. So we recognized we needed to activate lowercasing option.

4 Results

In this section, we show the resluts of the evaluations. First, we show the result of the evaluation with sentence clustering, and then the result of evaluation with fill mask task.

4.1 Result of Evaluation with Sentence Clustering

Table 2 summarizes A_m, B_m, and scores of evaluation with sentence clustering. Bigger B_m is better, and smaller A_m and score are better.

Table 2: Values and scores of evaluation with sentence clustering

Models	A_m	B_m	Score
Kyoto Univ. Ver.	240131.79	337.83	710.81
MeCab Ver.	97536.21	154.37	631.06
SP Ver.	67744.36	104.05	651.06
Tohoku Univ. Ver.	49991.31	65.64	761.58
NICT Ver.	106698.11	151.27	705.37
Laboro Ver.	153378.22	273.83	**560.13**

The following shows the results of comparing models by score, A_m, and B_m, and figure 2 is a bar graph of the results.

Score: Laboro Ver. < MeCab Ver. < SP Ver. < NICT Ver. < Kyoto Univ. Ver. < Tohoku Univ.Ver.
A_m: Tohoku Univ. Ver. < SP Ver. < MeCab Ver. < NICT Ver. < Laboro Ver. < Kyoto Univ. Ver.
B_m : Kyoto Univ. Ver. > Laboro Ver. > MeCab Ver. > NICT Ver. > SP Ver. > Tohoku Univ.Ver.

Table 3 shows scores of models except NICT Ver., and Laboro Ver. in previous work (Shibayama et al., 2020) and this work. According to this table, Scores of models except MeCab Ver. changes about 0.8–30 from results in previous evaluation (Shibayama et al., 2020). These changes did not influence the results of comparisons. Score of MeCab Ver. model became 100 or more higher than the previous result, but this change also did not influence results.

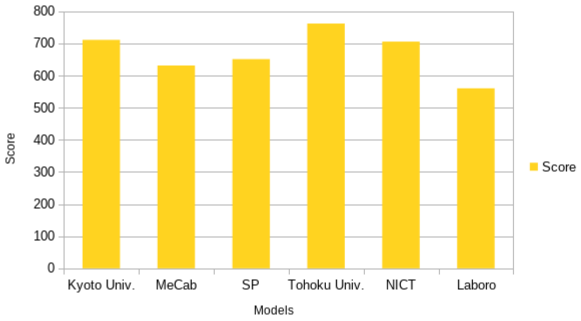

Figure 2: The results of comparing models by score

Table 3: Scores of previous work(Shibayama et al., 2020) and this work

Models	previous	this
Kyoto Univ. Ver.	710.88	710.81
MeCab Ver.	458.19	631.06
SP Ver.	668.92	651.06
Tohoku Univ. Ver.	792.34	761.58

4.2 Result of Re-evaluation with Fill Mask Task

Table 4 shows average of percentages that mask token is the word in matching label of all domains and three domains: books, DVDs, and music. Figure 3 shows a bar graph of column “All” in table 4.

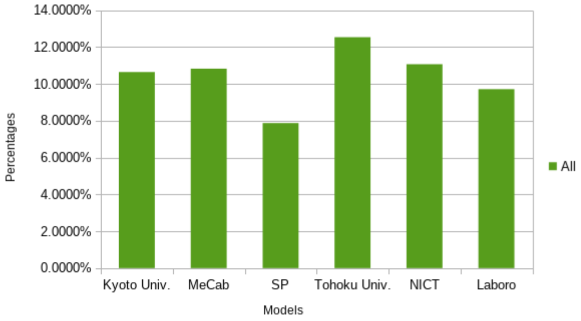

Figure 3: Averages of percentages of all domains

The following shows the result of comparing models by percentages of all domains, and this is different from the results in Section 4.1.

Table 4: Average of percentages that mask token is true masked word

Models	books	DVDs	music	All
Kyoto Univ. Ver.	11.53%	11.18%	9.24%	10.65%
MeCab Ver.	11.24%	13.62%	7.62%	10.83%
SP Ver.	7.36%	9.86%	6.41%	7.88%
Tohoku Univ. Ver.	14.04%	12.76%	10.81%	**12.54%**
NICT Ver.	11.90%	12.63%	8.68%	11.07%
Laboro Ver.	8.86%	10.44%	9.85%	9.72%

Tohoku Univ. Ver. > NICT Ver. > MeCab Ver. > Kyoto Univ. Ver. > Laboro Ver. > SP Ver.

5 Discussion

In this section, we describe the results in Section 4, and why there is a difference between the results in Section 4.1 and Section 4.2.

We changed the tokenizer settings for MeCab Ver. model from previous evaluation not to use subword tokenize[10]. We think this influenced the score of MeCab Ver. model, which caused a difference from a previous work (Shibayama et al., 2020).

As mentioned earlier, we considered A_m and B_m as in-class and out-of-class dispersion, respectively, in order to calculate easily (see, footnotes of Section 3.1). Therefore, comparing A_m means evaluating whether embeddings in the same class are close, and B_m means evaluating differences of embeddings that are not in the same class. We can dedude the general tendencies of each model from the results in Section 4.1. The best model is Laboro Ver., which has the second-highest B_m and about 100000 smaller A_m than Kyoto Univ. Ver. model. MeCab Ver. model that has the best score in previous eval-

[10] According to an article of MeCab Ver. model, we have to change scripts that use only MeCab as a tokenizer.

uation (Shibayama et al., 2020) is the second-best model. SP Ver. model is the third, which A_m of model is it of Tohoku Univ.Ver. model or more and it of MeCab Ver. model less. Tohoku Univ. Ver. model has the worst score, which has smallest A_m and B_m. This means the dispersion of all embeddings is smaller than the other models.

However, the results in Section 4.1 are different from the results in Section 4.2. Thus, we could not conclude that the results of methods of evaluation with sentence clustering and fill mask task have the same tendency. We used the title of articles in evaluation with sentence clustering, but we used a sentence in product reviews (see synopsis of Webis-CLS-10 (Prettenhofer and Stein, 2010)) with fill mask task. This difference may have caused the differences between the results in Section 4.1 and Section 4.2.

6 Conclusion

We evaluated Japanese pre-trained BERT models using sentences that were labeled, and outputs of BERT that inputted those sentences. Then, we obtained the following result.

> Laboro Ver. < MeCab Ver. < SP Ver. < NICT Ver. < Kyoto Univ. Ver. < Tohoku Univ. Ver.

Also, we masked a specific noun in each sentence, calculated percentage that mask token is the word in matching label, and re-evaluated with averages of that percentage. However, we obtained the following result, and this is different from result of sentence clustering.

> Tohoku Univ. Ver. > NICT Ver. > MeCab Ver. > Kyoto Univ. Ver > Laboro Ver. > SP Ver.

If we decrease the difference of type or domain of documents that are used in both experiments, there is a chance that the comparison results will be different from what we obtained in this work.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP19K12093.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-2019*, pages 4171–4186.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS-2013*, pages 3111–3119.

Peter Prettenhofer and Benno Stein. 2010. Cross-Language Text Classification using Structural Correspondence Learning. In *48th Annual Meeting of the Association of Computational Linguistics (ACL 10)*, pages 1118–1127. Association for Computational Linguistics, July.

Yuuya Sakaizawa and Mamoru Komachi. 2016. Building similarity dataset of japanese verbs and adjectives (in Japanese). *The Twenty-second Annual Meeting of the Association for Natural Language Processing*, pages 258–261.

Naoki Shibayama, Rui Cao, Jing Bai, Wen Ma, and Hiroyuki Shinnou. 2019. A comparison of japanese pretrained bert models (in Japanese). *IEICE Techn. Rep.*, 119(212):89–92.

Naoki Shibayama, Rui Cao, Jing Bai, Wen Ma, and Hiroyuki Shinnou. 2020. Evaluation of pretrained BERT model by using sentence clustering (in Japanese). In *The Twenty-sixth Annual Meeting of the Association for Natural Language Processing*, pages 1233–1236.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

A Basic Hyperparameters of Models We Evaluated

In this section, we show basic hyperparameters of pre-trained BERT models we evaluated. However, some parameters were not written in both config file of model and model publisher's web site. Table 5 shows basic pre-training information of pre-trained models. "No Info" cell is a parameter that we could not found the correct value. Some publishers pre-trained the model with two step pre-training, and we show those as Ph1 parameter and Ph2 parameter if there is differences.

Table 5: Basic pre-training information of BERT models

Models	Model Size	Whole Word Masking	Vocabulary Size	max_ seq_ length
Kyoto Univ. Ver.	Base	No	32,000	128
MeCab Ver.	Base	No	32,000	No Info
SP Ver.	Base	No	32,000	No Info
Tohoku Uinv. Ver.	Base	No	32,000	512
NICT Ver.	Base	No	32,000	Ph1-128 Ph2-512
Laboro Ver.	Base	No	32,000	Ph1-128 Ph2-512