Construction and Evaluation of Japanese Sentence-BERT Models

Naoki Shibayama Hiroyuki Shinnou
Ibaraki University, Ibaraki, Japan
{21nd303a, hiroyuki.shinnou.0828}@vc.ibaraki.ac.jp

Abstract

Sentence-BERT is model which based on "Bidirectional Encoder Representations from Transformers" (BERT) for building sentence embedding. This model has abilities for semantic analysis similar to BERT; however, the processing need not be online like in BERT. Therefore, it is also effective for similar sentence searches. No Japanese sentence-BERT model has been released in the right format. Here, we built six Japanese sentence-BERT models with Japanese Stanford natural language inference (JSNLI) released at Kyoto University and six public Japanese BERT models. Furthermore, we proposed two evaluation methods for the sentence-BERT models and evaluated the six Japanese sentence-BERT models using the ratio of the in-class dispersion to out-of-class dispersions and accuracy of classification tasks using k-nearest neighbor (k-NN) classifier. As a result, two sentence-BERT models recorded higher performance: the model which was built from Tohoku BERT and the National Institute of Information and Communications Technology (NICT) BERT.

1 Introduction

Sentence-BERT (Reimers and Gurevych, 2019) is a specialized BERT (Devlin et al., 2018) for building sentence embedding. Normal sentence-BERT builds embedding for the input sequences of token IDs using the averages of output embeddings from BERT. The best feature of sentence-BERT is that the processing need not be online. A similar sentence search task requires suitable sentence embedding to measure similarity of two sentences; however, embeddings made from simple models, such as a bag of words, cannot measure the similarity of meanings. While BERT overcomes this drawback, it can process cross-encoders to solve a task requiring a pair of sentences. Although BERT is not used from a viewpoint of processing time, the sentence-BERT processes as a bi-encoder; therefore, the problem of processing time, as in the case of BERT, does not arise. Sentence-BERT can be also used for tasks which requires similarity of sentence meanings, and sentence-BERT has high effectiveness (Ndukwe et al., 2020)(Shirafuji et al., 2020)(Gencoglu, 2020).

However, there is no proper pretrained model for Japanese sentence-BERT. Therefore, we propose a Japanese sentence-BERT that requires a Japanese-based pretrained BERT and large natural language inference (NLI) dataset. Here, we use the Japanse Stanford natural language inference (JSNLI) releaseatby Kyoto University as our large NLI dataset. Furthermore, we used six Japanese pretrained BERT to build and evaluate six sentence-BERTs: KyotoUniv. BERT, Stockmark BERT, Sentence-Piece BERT, Tohoku BERT, National Institute of Information and Communications Technology (NICT) BERT, and Laboro BERT.

Evaluating the sentence-BERT requires setting tasks needed for the sentence-BERT and measuring the performances of the tasks. The appropriate tasks for evaluating sentence-BERT are tasks such as searching similar sentence and judging similarity between two sentences. However, cases suffice where preparing datasets of those tasks are impossible. In this work, we devised two evaluation methods to evaluate sentence-BERT. One of the methods embeds the title sentences of each document from dataset for document classification, uses labels as classes, and measures the in-class and out-of-class dispersion. The other embeds each sentence from the dataset for sentence positive / negative judging and measures the accuracy of identification with k-nearest neighbor (k-NN) classifier. Both methods can evaluate the suitability of sentence embedding and the sentence-BERT.

The results of the experiments show that the Tohoku BERT has a high performance, followed by NICT BERT.

2 Related works

Research on building embedding for sentences remains active, and many methods are proposed until now.

Paragraph Vector (Le and Mikolov, 2014)[1] is the model that extended word2vec (Mikolov et al., 2013) to documents: word2vec outputs word embedding as vectors. When a sentence is input into a trained model, a sentence embedding can be obtained. Skip-Thought (Kiros et al., 2015) which is Encoder-Decoder model is also the method to build sentence embedding. It tokenizes sentences to obtain word vectors, inputs these vectors the pretrained encoder, and builds the sentence embedding.

Recently, a universal sentence encoder (Cer et al., 2018) and sentence-BERT were proposed. The universal sentence encoder is specialized encoder for outputting sentence embedding. Two architectures were proposed: one requires low computational resources, and the other one uses transformer, and performance is valued. The sentence-BERT inherits some features from second architecture: the BERT is a transformer-based model.

Many research uses sentence-BERT or embedding as outputs. Ifeanyi et al. (Ndukwe et al., 2020) researched automatic grading to three types of questions which required short answer with pretrained sentence-BERT. Shirafuji et al. (Shirafuji et al., 2020) trained Japanese sentence-BERT which was specialized for politics and used it to summarize proceedings. Gencoglu Oguzhan (Gencoglu, 2020) used embedding that pretrained sentence-BERT outputs to detect the side effects of drugs.

3 Building sentence-BERT

In this section, we state about how to build and use sentence-BERT.

sentence-BERT is built with following steps: A pooling layer is added to the pretrained BERT and trained with NLI tasks. There are three basic pooling methods: using the [CLS] token, using the token which is the biggest token, and using the average of all tokens. The building steps for the sentence-BERT are the same, irrespective of the pooling method. Finally, building a sentence-BERT requires preparing a NLI dataset and the pretrained BERT model.

In this work, we used Japanese SNLI dataset (Yoshikoshi et al., 2020)[2] which was released by Kyoto University. This dataset was collected as follows: First, the machine translates The Stanford NLI (SNLI) corpus (Bowman et al., 2015) into Japanese and filters train data and validation data using BLEU score and crowdsourcing, respectively. This dataset also includes nonfiltered training data. We used filtered training data as training data in this work. We validated the model using validation data when each epoch was finished and selected and saved the best sentence-BERT. Table 1 shows the data size of this dataset.

The Japanese sentence-BERT in this work re-

[1] This model is also called doc2vec.

[2] https://nlp.ist.i.kyoto-u.ac.jp/index.php?日本語SNLI(JSNLI)データセット

Table 1: Size of Japanese SNLI dataset

	Nonfiltered training data	Filtered training data	Validation data
Number of pairs	548,014	533,005	3,916

quired the pretrained BERT model to pretrain with Japanese dataset. We built six sentence-BERTs with six models from Japanese pre-trained BERT, which was released by Kyoto University (KyotoUniv. BERT)[3], Morinaga (Stockmark BERT)[4], Yohei Kikuta on GitHub (Sentence-Piece BERT)[5], Tohoku University (Tohoku BERT)[6], NICT (NICT BERT)[7], and Laboro.AI Inc. (Laboro BERT)[8]. Table 2 shows the features of each model. We used models that represent a token as a word and conducted sub-word tokenization of different available versions.

Here, we used "Sentence-Transformers"[9] to build sentence-BERT. Creating sentence-BERT from "Huggingface Transformers" (Wolf et al., 2019) BERT models with this library is easy. Following and figure 1 shows summary of method. More information can be obtained in document pages of Sentence-Transformers[10].

1. Import "sentence_transformers".
2. Load BERT model with "sentence_trasnformers.models.Transformer" class.

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

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

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

[6]https://github.com/cl-tohoku/bert-japanese We used bert-base-japanese.

[7]https://alaginrc.nict.go.jp/nict-bert/index.html We used BPE version.

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

[9]https://www.sbert.net/index.html

[10]https://www.sbert.net/docs/training/overview.html

Table 2: Features of Japanese pretrained BERT models

Model	Model size	Tokenizer	Training corpus
KyotoUniv. BERT	Base	Juman++	Japanese Wikipedia
Stockmark BERT	Base	MeCab (NE-ologd)	Japanese articles of business news
Sentence-Piece BERT	Base	Sentence-Piece	Japanese Wikipedia
Tohoku BERT	Base	MeCab	Japanese Wikipedia
NICT BERT	Base	MeCab (Ju-mandic)	Japanese Wikipedia
Laboro BERT	Base	Sentence-Piece	Text on the internet (12GB)

3. Prepare a pooling layer with "sentence_transformers.models.Pooling" class and set the pooling method with arguments of the class.

4. Prepare sentence-BERT with "sentence_transformers.SentenceTransformer" class and set the list arranged in the order of BERT pooling layer to argument "modules".

5. Select loss function (validation method) from "sentence_transformers.losses (evaluation)" which fits to train (validation) data.

6. Start learning with "fit" method of the sentence-BERT.

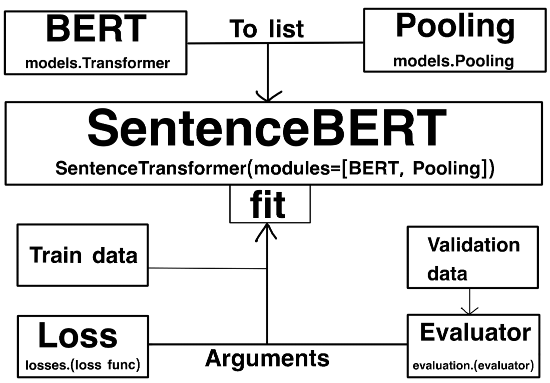

Figure 1: Abstract of making sentence-BERT

Obtaining sentence embedding with sentence-BERT requires one to preprocess[11] and input to encode method of the model. Depending on base BERT model, the same preprocessing can be used to train sentence-BERT.

In this work, we trained and evaluated the sentence-BERT using the Japanese SNLI training data (filtered) and validation data. The evaluation function was used when each epoch was finished, and the model with the highest score was saved. Table 3 shows the parameters for training. We used the default value of the Sentence-Transformer for parameters not presented in the table. We used "SoftmaxLoss" class as loss function and "LabelAccuracyEvaluator" class as evaluation function.

Table 3: Parameters to build sentence-BERT

Epochs	Batch size
20	16

4 Evaluation methods for sentence embedding

In this section, we present the evaluation methods for sentence embedding using sentence-BERT, which was achieved in the previous section.

4.1 Evaluation with ratio of the in-class and out-of-class dispersion

In this work, two methods were applied to evaluate the embedding. The first method is the ratio of the in-class and out-of-class dispersion. We previously proposed this method as evaluation method for Japanese pre-trained BERT model (Shibayama et al., 2020a)(Shibayama et al., 2020b). The method is outlined in the following steps and figure 2.

1. Prepare dataset, including labeled sentences.

2. Input sentences into target model and obtain embedding.

3. Classify embedding with labels that were given to the original sentences of the embedding.

4. Calculate the centroids of each class and average of in-class dispersion.

5. Calculate the average of centroids and average of out-of-class dispersion.

6. Divide average of the in-class dispersion by average of the out-of-class dispersion and use it as the score of target model.

[11] Some models use tokenizer for Japanese, but other models use default tokenizer that cannot be used for Japanese. Other models tokenized sentences with the following steps: preprocess sentences using a software or a tokenizer that models use when pretraining, and tokenize those with default tokenizer.

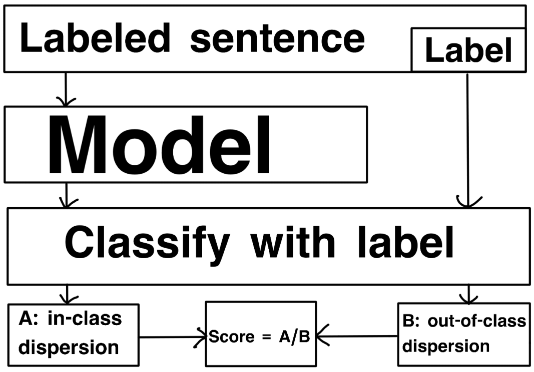

Figure 2: Evaluation method with ratio of the in-class and out-of-class dispersion

We used the method in previous works which parts of calculations were simplified (evaluation method with document clustering). The square of deviation was used as dispersion, and the sums of the in-class and out-of-class dispersion were used as averages in this method. This simplifies the calculation, but the number of embedding of each class must be the same. We used parts of Livedoor news corpus[12] as a dataset with labeled sentences, which satisfied this precondition. This corpus includes nine categories of news articles and that is nonlabeled dataset, but we used "dataset with article titles which extracted one hundred per each category from that dataset and were given category label" (Shibayama et al., 2020a)(Shibayama et al., 2020b), made in previous works.

4.2 Evaluation method with sentence classification by k-NN classifier

We used simple classifier as the second evaluation method: We calculated the classification task for sentences with a simple classifier that uses k-NN and compared the accuracy. In this work, we used Tsukuba sentiment-tagged corpus (TSUKUBA corpus) (Rakuten Group, 2014). This dataset is review data of Rakuten Travel informeation that has sentiment-tag per each sentence. We used 2672 sentences of this dataset, splitting 80% to the training data, and 20% to the test data. Table 4 shows the information on the data used.

Table 4: Label distribution of TSUKUBA corpus we used

	Training data	Test data	The sum
Label P	1,476	369	1,845
Label K	662	165	827
The sum	2,138	534	2,672

We input the training and test data to each model, trained 5-NN classifier with train embedding, and collected and compared the accuracy of the test embedding.

5 Results

Section 1 mentioned that no right pretrained model existed, not that there was no Japanese pretrained sentence-BERT model. The Japanese pretrained sentence-BERT[13] was released by Isamu Sonobe from NS Solutions Corporation. This model is based on the Tohoku BERT; however, no information of pretraining corpus and hyperparameters is provided. We denote this model as SBERT-jp in this work. We also selected the results of this model as the baseline, and compared other sentence-BERT models built from BERT.

Table 5 and figure 3 show the results of the ratio of the in-class and out-of-class dispersion. A_m and B_m are the averages of the in-class and out-of-class dispersions mentioned in section 4.1, and lower score means that better sentence embedding are created.

Table 6 and figure 4 show results of the evaluation with 5-NN classifier.

In evaluation using document clustering, Tohoku SBERT obtained the same score as NICT SBERT, followed by KyotoUniv. SBERT. We obtained the same pattern for Tohoku and NICT SBERT in the result of the evaluation with 5-NN classifier. However, the rank of Laboro and KyotoUniv. SBERTs

[12] https://www.rondhuit.com/download.html#ldcc

[13] https://qiita.com/sonoisa/items/1df94d0a98cd4f209051

Table 5: Result of the evaluation with document clustering

Model	A_m	B_m	Score
SBERT-jp	222,138.34	212.94	1,043.19
KyotoUniv. SBERT	204,566.35	253.74	806.20
Stockmark SBERT	230,969.95	103.85	2,224.04
SP SBERT	310,377.36	237.53	1,306.70
Tohoku SBERT	179,802.27	252.77	711.32
NICT SBERT	197,759.28	268.30	737.10
Laboro SBERT	211,038.32	222.27	949.48

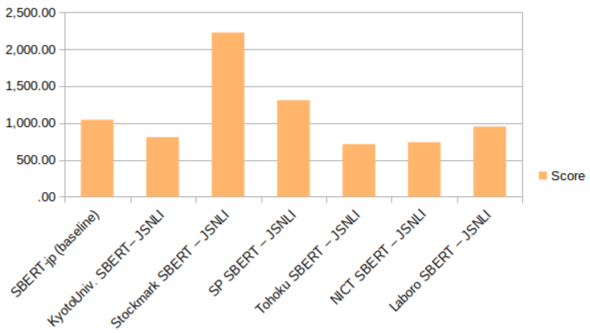

Figure 3: Result of the evaluation with document clustering (graph)

Table 6: Result of the evaluation with 5-NN classifier (%)

Model	Score
SBERT-jp	94.19
KyotoUniv. SBERT	85.96
Stockmark SBERT	76.97
SP SBERT	81.84
Tohoku SBERT	90.64
NICT SBERT	89.33
Laboro SBERT	86.33

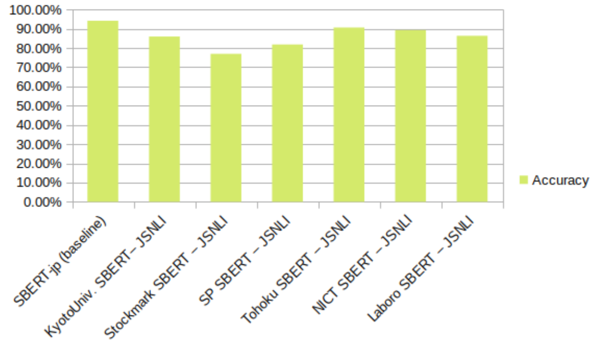

Figure 4: Result of the evaluation with 5-NN classifier (graph)

changed. There are models better than the baseline in the evaluation with document clustering, but no models better than the baseline in the evaluation with 5-NN clustering.

We obtained this conclusion from the experimental results: Tohoku BERT and NICT BERT are good when we build sentence-BERT with Japanese-released pretrained BERT.

6 Discussion

6.1 Fine-tuning sentence-BERT

Sentence-BERT can build better sentence embedding than did BERT. However, there is possibility that the embedding are not good for fine-tuning. We evaluated this with document classification task in this section.

We used Livedoor news corpus in this experiment. We used article titles of this corpus in previous section but use main parts of the articles this time. This corpus contains 7,376 articles belonging to nine categories. We shuffled this dataset and divided it into 10 equal sets. We then selected one training and test data. We also classified documents with feature-based using of BERT or sentence-BERT: we fixed the parameters of BERT and trained only classification layer. We trained 20 epochs with training data, and measured the accuracy of the test data when each epoch was completed.

Figure 5 shows the result of comparison between

Tohoku BERT and sentence-BERT.

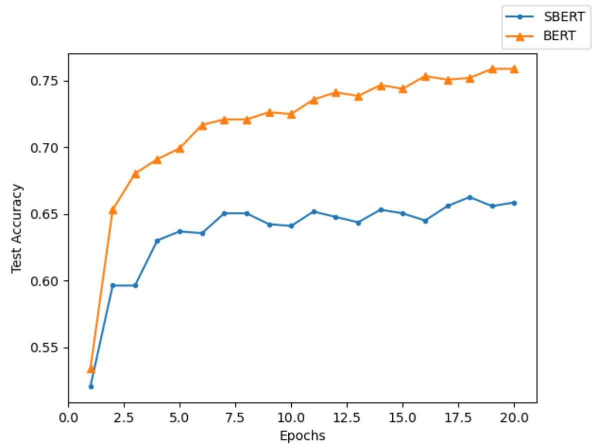

Figure 5: Accuracy comparing between feature based Tohoku BERT and SBERT

Figure 6 shows the result between NICT BERT and sentence-BERT.

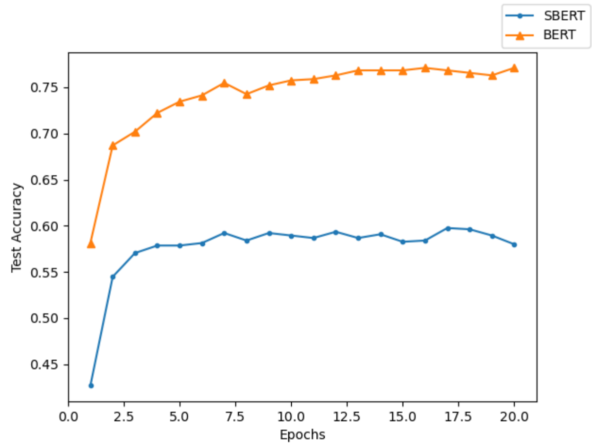

Figure 6: Accuracy comparing between feature based NICT BERT and SBERT

Figure 5 and 6 showed that BERT obtained higher accuracy than did sentence-BERT, and the differences gradually become larger as the training continues.

These show that sentence embedding, sentence-BERT outputs, does not always fit to the networks of classifiers. Also, fine-tuning is ineffective for sentence-BERT. We have to find how fine-tuning for sentence-BERT works better.

6.2 Using for one-shot learning

We are considering about one-shot learning as one application of sentence-BERT. If sentence-BERT builds better sentence embedding, we expect high performance for sentence classification with small training data. We evaluated this point using the TSUKUBA corpus which used in section 5. First, we randomly select one positive (label P) and negative (label K) sentence from the training data and translated these to sentence embedding using sentence-BERT. These are used as training data. Further, we translated the test data to embedding with sentence-BERT. We measured sentence classification accuracy for the test dataset using the nearest neighbor algorithm (1-NN). The above experiment was performed five times with both Tohoku and NICT sentence-BERT. Table 7 shows the results of the experiment.

Still there are variations, the Tohoku sentence-BERT achieved approximately 0.7 accuracy. Since we randomly selected the training data, this result has high accuracy. We also think that the performance can be improved by changing the methods for selecting training data and using unlabeled data. Furthermore, using the sentence-BERT for one-shot learning can be considered.

7 Conclusion

We introduced, discussed, and presented the results of the methods for building and evaluating six sentence-BERT models. We used JSNLI released by Kyoto University and six Japanese pretrained BERT. We also proposed two methods for evaluating the sentence-BERT: method using the ratio of the in-class and out-of-class dispersion, and the method evaluating the accuracy for sentence classification task using a k-NN simple classifier. The results of the proposed methods showed that the Tohoku BERT has high performance as the base of sentence-BERT, followed by the NICT BERT. The effect of fine-tuning the sentence-BERT and the usability of sentence-BERT for one-shot learning is open for fur-

Table 7: One-shot learning accuracy with sentence-BERT (%)

Model	1st	2nd	3rd	4th	5th	Average
Tohoku SBERT	77.49	75.05	45.78	78.61	79.55	71.30
NICT SBERT	61.54	51.78	46.53	73.17	82.55	63.11

ther explorations.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP19K12093 and ROIS NII Open Collaborative Research 2021(2021-FC05). The authors would like to thank the Enago (`www.enago.jp`) for the English language review.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Oguzhan Gencoglu. 2020. Sentence transformers and bayesian optimization for adverse drug effect detection from twitter. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 161–164.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *arXiv preprint arXiv:1506.06726*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 22–24 Jun. PMLR.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Ifeanyi G Ndukwe, Chukwudi E Amadi, Larian M Nkomo, and Ben K Daniel. 2020. Automatic grading system using sentence-bert network. In *International Conference on Artificial Intelligence in Education*, pages 224–227. Springer.

Inc Rakuten Group. 2014. Tsukuba sentiment-tagged corpus. text(tsv).

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Naoki Shibayama, Rui Cao, Jing Bai, Wen Ma, and Hiroyuki Shinnou. 2020a. Evaluation of pre-trained bert model with sentence clustering (japanese paper). In *The Twenty-sixth Annual Meeting of the Association for Natural Language Processing*, pages 1233–1236.

Naoki Shibayama, Rui Cao, Jing Bai, Wen Ma, and Hiroyuki Shinnou. 2020b. Evaluation of pretrained bert model by using sentence clustering. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 279–285, Hanoi, Vietnam, 10. Association for Computational Linguistics.

Daiki Shirafuji, Hiromichi Kameya, Rafal Rzepka, and Kenji Araki. 2020. Summarizing utterances from japanese assembly minutes using political sentence-bert-based method for qa lab-poliinfo-2 task of ntcir-15. *arXiv preprint arXiv:2010.12077*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Takumi Yoshikoshi, Daisuke Kawahara, Sadao Kurohashi, et al. 2020. Multilingualization of a natural language inference dataset using machine translation (japanese paper). *Technical Reports: Natural Language Processing (NL)*, 2020(6):1–8.