

# 日本語 SentenceBERT の構築とその評価

芝山 直希<sup>1,a)</sup> 新納 浩幸<sup>2,b)</sup>

概要：SentenceBERT は文の埋め込み表現の構築に特化した BERT である。BERT のように意味の解析能力を有している一方、BERT とは異なりオンラインで実行する必要がないため、類似文検索などでの有用性が高い。しかし、現在、適切な形で公開されている日本語 SentenceBERT モデルは存在しない。そこでここでは京都大学で公開されている日本語版 SNLI (JSNLI) と公開されている日本語 BERT モデル 6 つを用いて 6 つの日本語 SentenceBERT を構築した。また SentenceBERT モデルの評価方法として、クラス内分散とクラス間分散の比を測る方法、及び k-NN 法による単純な分類器での文の分類問題に対する精度を用いる方法を提案し、作成した 6 つの日本語 SentenceBERT を評価した。結果、東北大版 BERT と NICT 版 BERT から構築した SentenceBERT が同程度の高い性能を示した。

キーワード：SentenceBERT, k-means, クラスタ内分散, クラスタ間分散, k-NN

## Construction and Evaluation of Japanese Sentence-BERT Models

SHIBAYAMA NAOKI<sup>1,a)</sup> SHINNOU HIROYUKI<sup>2,b)</sup>

### 1. はじめに

我々は日本語 SentenceBERT[9] の構築を行っている。京都大学で公開されている日本語版 SNLI (JSNLI) と 6 つの日本語 BERT[3] を用いて、6 つの SentenceBERT を構築した。またそれらを独自の評価法により評価した。結果、東北大版 BERT と NICT 版 BERT を用いて構築した SentenceBERT が同程度に性能が高かった。

SentenceBERT は文の埋め込み表現の構築に特化した BERT である。標準的な SentenceBERT は、入力 token 列に対して内包する BERT が出力する token の埋め込み表現列の平均ベクトルを取ることで、入力文に対する埋め込み表現を構築する。SentenceBERT の最大の特徴はオンラインで SentenceBERT を動かす必要がない点である。

類似文検索のタスクでは 2 文間の類似度を測るために、適切な文の埋め込み表現を必要とするが、Bag of Words などの簡易なモデルから作られる埋め込み表現では、意味的な類似度を測るには力不足である。またそのような用途に応えられる埋め込み表現を作るために、BERT を利用することも考えられるが、2 文入力タイプのタスクに対して BERT は Cross Encoder の処理をとるため、処理時間の観点から、実質、利用できない。一方、SentenceBERT は Bi-Encoder の処理であるため、2 文入力タイプのタスクに対しても BERT のような処理時間の問題を受けず、上記用途に利用できる。他にも意味的な文の類似度を必要とするタスクに SentenceBERT は利用できるため、その有用性は高い [8][11][4]。

しかし日本語の SentenceBERT に対しては、適当な学習済みモデルが公開されていないため、我々は日本語の SentenceBERT を構築を行っている。SentenceBERT を構築する場合、基になる日本語 BERT の他に大規模な自然言語推論のデータセットが必要である。ここでは大規模な自然言語推論のデータセットとして、京都大学で公開されている JSNLI 用いる。また日本語 BERT として

<sup>1</sup> 茨城大学大学院理工学研究科情報工学専攻  
Major in Computer and Information Sciences, Graduate  
School of Science and Engineering, Ibaraki University

<sup>2</sup> 茨城大学大学院理工学研究科情報科学領域  
Graduate School of Science and Engineering, Department of  
Computer and Information Sciences, Ibaraki University

a) 21nd303a@vc.ibaraki.ac.jp

b) hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

(1) 京大版 BERT (2) Stockmark 社版 BERT (3) SentencePiece 版 BERT (4) 東北大版 BERT (5) NICT 版 BERT (6) Laboro 社版 BERT の 6 つのモデルをそれぞれ利用し、6 つの SentenceBERT を構築し、それらを評価した。

SentenceBERT を評価する場合、実際に SentenceBERT を必要とするタスクを設定し、そのタスクに対する性能評価により評価するのが一般的である。SentenceBERT に関しては類似文検索や 2 文間の類似度判定のタスクなどが適切なタスクと考えられるが、そのようなデータセットが用意できない場合は多い。ここでは SentenceBERT の評価を行うために 2 つの評価方法を考案した。一つは文書分類のデータセットから SentenceBERT を用いて各文書のタイトル文の埋め込み表現を作成し、ラベルをクラスとしたクラス内分散、クラス間分散を測る方法である。もう一つは文のポジネガ判定のデータセットから SentenceBERT を用いて各文の埋め込み表現を作成し、k-NN を用いて識別の精度を測る方法である。どちらの方法においても文の埋め込み表現の適切さを評価することができるため、利用した SentenceBERT を評価できる。

実験の結果、東北大版 BERT と NICT 版 BERT から構築した SentenceBERT が同程度の高い性能を示した。

## 2. 関連研究

文に対する埋め込み表現の構築に関する研究は活発であり、これまで多くの手法が提案されている。

Paragraph Vector[6]<sup>\*1</sup> は単語の分散表現を数値ベクトルとして出力する word2vec[7] を、文章に対応できるように拡張したモデルである。1 文を訓練後のモデルに入力すれば文の埋め込み表現が得られる。Encoder-Decoder 型のモデルである Skip-Thought[5] も文の埋め込み表現を作成する手法である。文を事前に tokenize し単語ベクトル化した後に、事前学習済みの Encoder に入力することで文の埋め込み表現を構築する。

近年では Universal Sentence Encoder[2] や本研究の核である SentenceBERT が提案されている。Universal Sentence Encoder は文の埋め込み表現の出力に特化した Encoder である。必要計算資源量の少ないアーキテクチャと Transformer を採用した性能重視のアーキテクチャの二種類が提案されており、Transformer をベースとする BERT を採用した SentenceBERT は後者の特徴を一部継承している。

SentenceBERT や SentenceBERT が出力する埋め込み表現を利用した研究は多い。Ifeanyi ら [8] は事前学習済みの SentenceBERT を用いて、短い解答を要求する 3 種のバリエーションからなる設問群に対し、その解答を自動採点するシステムの研究を行った。Shirafuji ら [11] は日本語で

<sup>\*1</sup> doc2vec と呼ばれることもある。

事前学習された政治分野に特化した SentenceBERT を訓練し、議事録の要約に使用した。Oguzhan 氏 [4] は Twitter 上からの薬物の副作用に関する言及の検出に、学習済み SentenceBERT が出力する埋め込み表現を利用した。

## 3. SentenceBERT の構築

本節では SentenceBERT の構築・利用方法について述べる。

SentenceBERT は事前学習済みの BERT に Pooling 層を加え、自然言語推論タスクで追加学習することで構築される。Pooling の手法としては [CLS] token を使用する、最大の token を使用する、全 token の平均値を用いるという 3 つが基本となる。どの Pooling 手法を用いる場合でも、構築手法は同じである。結局、SentenceBERT の構築には自然言語推論のデータセットと事前学習済み BERT モデルの 2 つが必要となる。

本研究では自然言語推論のデータセットとして、京都大学で公開されている日本語 SNLI データセット [14]<sup>\*2</sup> を使用した。このデータセットは The Stanford Natural Language Inference (SNLI) Corpus[1] に対し機械翻訳を適用することで日本語化し、訓練データを BLEU スコアで、評価用データをクラウドソーシングによる人手でフィルタリングして作成されたものである。また、訓練データに関してはフィルタリングを適用する前の訓練データも同梱されている。本研究では学習データとしてフィルタリング適用後の訓練データを使用した。また、各 epoch 終了時に評価用データで Validation を行い、最良の SentenceBERT を選択・保存した。表 1 に収録データ数を示す。

表 1 日本語 SNLI データセットのサイズ

|     | 訓練データ (フィルタリング前) | 訓練データ (フィルタリング後) | 評価用データ |
|-----|------------------|------------------|--------|
| ペア数 | 548,014          | 533,005          | 3,916  |

本研究では日本語の SentenceBERT を構築するため、使用する事前学習済みの BERT も日本語のものが必要となる。本研究ではいくつか公開されている日本語の BERT モデルのうち、京都大学で公開されている BERT (京大版 BERT)<sup>\*3</sup>、Stockmark 社の森長氏が公開している BERT (Stockmark 社版 BERT)<sup>\*4</sup>、Yohei Kikuta 氏が GitHub 上で公開している BERT (SentencePiece 版 BERT)<sup>\*5</sup>、東北大学で公開されている BERT (東北大版 BERT)<sup>\*6</sup>、情報通信研究機構 (NICT) が公開している BERT (NICT

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

<sup>\*3</sup> <http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT日本語PretrainedモデルBASE通常版>を使用した。

<sup>\*4</sup> <https://qiita.com/mkt3/items/3c1278339ff1bcc0187f>

<sup>\*5</sup> <https://github.com/yoheikikuta/bert-japanese>

<sup>\*6</sup> <https://github.com/cl-tohoku/bert-japanese> bert-base-japanese モデルを用いた。

版 BERT)\*<sup>7</sup>, 株式会社 Laboro.AI が公開している BERT (Laboro 社版 BERT)\*<sup>8</sup> の 6 つの 事前学習済み日本語 BERT モデルを用いて SentenceBERT を構築した. それぞれの特徴を表 2 に示す. なお, これらのモデルはいずれも単語単位で token 化を行うものを用いた.

表 2 各事前学習済み日本語 BERT の特徴

| モデル                  | モデルサイズ | Tokenizer        | 学習用コーパス              |
|----------------------|--------|------------------|----------------------|
| 京大版 BERT             | base   | Juman++          | 日本語 Wikipedia        |
| Stockmark 社版 BERT    | base   | MeCab (NEologd)  | 日本語ビジネスニュース記事        |
| SentencePiece 版 BERT | base   | SentencePiece    | 日本語 Wikipedia        |
| 東北大版 BERT            | base   | MeCab            | 日本語 Wikipedia        |
| NICT 版 BERT          | base   | MeCab (Jumandic) | 日本語 Wikipedia        |
| Laboro 社版 BERT       | base   | SentencePiece    | インターネット上のテキスト (12GB) |

本研究では SentenceBERT の構築に SentenceTransformers\*<sup>9</sup> を使用した. 本ライブラリを用いることで, Huggingface Transformers[12] 形式で保存されている BERT モデルから SentenceBERT を容易に作成できる. その手順の概要を図 1 及び以下に示す. より詳細な手順は SentenceTransformers のドキュメント内\*<sup>10</sup>に示されている.

- (1) sentence\_transformers を import する.
- (2) sentence\_transformers.models.Transformer クラスを用いて BERT を読み込む.
- (3) sentence\_transformers.models.Pooling クラスで Pooling 層を作成し, Pooling 手法を設定する.
- (4) sentence\_transformers.SentenceTransformer クラスの引数 modules に, BERT, Pooling 層の順に並べられたリストを渡す.
- (5) 使用する学習データ (評価データ) に適した Loss 関数 (Evaluator) を sentence\_transformers.losses (evaluation) から選択する.
- (6) 準備した SentenceTransformer の fit メソッドで学習を開始する.

\*<sup>7</sup> <https://alaginrc.nict.go.jp/nict-bert/index.html> BPE ありのものを用いた.

\*<sup>8</sup> <https://laboro.ai/column/laboro-bert/>

\*<sup>9</sup> <https://www.sbert.net/index.html>

\*<sup>10</sup> <https://www.sbert.net/docs/training/overview.html>

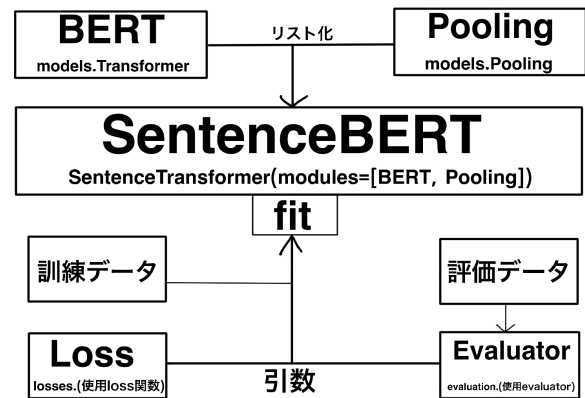


図 1 SentenceBERT の構築方法

ここで作成した SentenceBERT を用いて文の埋め込み表現を得たい場合, 必要に応じて前処理を行い\*<sup>11</sup>, encode メソッドに入力すれば良い. 元とする BERT モデルにもよるが, SentenceBERT の学習時にも同様の前処理が必要となる.

本研究では前述した日本語 SNLI 訓練用データセット (フィルタリング後) を用いて学習を行い, SentenceBERT を作成した. 各 epoch 終了時に日本語 SNLI 評価用データセットと評価関数を用いてモデルを評価し, 最高値を記録した epoch 終了時点のモデルを保存するようにした. 学習時のパラメータを表 3 に示す. 記述していないパラメータは前述したライブラリのデフォルト値を使用している. また, Loss 関数として SoftmaxLoss を, 評価関数として LabelAccuracyEvaluator を指定・利用した.

表 3 SentenceBERT 作成時のパラメータ

| epoch 数 | バッチサイズ |
|---------|--------|
| 20      | 16     |

#### 4. 文の埋め込み表現の評価方法

本節では前節で作成した SentenceBERT から得られる文の埋め込み表現を評価する方法について述べる.

##### 4.1 クラス内分散とクラス間分散の比を用いた評価

本研究では 2 つの評価手法を用いて埋め込み表現を評価した. そのうちの 1 つがクラス内分散とクラス間分散の比を用いた評価手法である. この手法は過去に事前学習済み日本語 BERT モデルの評価手法として我々が提案したものである [15][10]. その手順を図 2 及び以下に示す.

- (1) ラベル付き文からなるデータセットを用意する.

\*<sup>11</sup> 一部のモデルは日本語専用 Tokenizer を用いているが, BERT 標準の日本語に対応していない Tokenizer を用いているモデルも多い. 後者のモデルはモデル毎に指定されたソフトウェアや Tokenizer で前処理した文を, モデルの Tokenizer で更に Tokenize する方式を用いている.

- (2) 評価対象となるモデルに文を入力し、埋め込み表現を得る。
- (3) 埋め込み表現の元の文に付与されたラベルを用いて、埋め込み表現をクラス分けする。
- (4) 各クラスの重心を導出し、クラス内分散の平均を算出する。
- (5) 全クラスの重心の平均を計算し、クラス間分散の平均を求める。
- (6) クラス内分散の平均をクラス間分散の平均で割り、その値を評価対象の評価値とする。



図 2 クラス内分散・クラス間分散の比を用いた評価方法

本研究では当該研究で示されている計算の一部が簡略化された手法(文書クラスタリングによる評価手法)を採用した。この手法では偏差の2乗を分散とみなし、クラス内・クラス間分散の双方において総和を平均の代替として使用した。これにより計算が簡略化されているが、各クラスに属する埋め込み表現の数が等しいことが前提となる。この前提を満たすラベルが付与された文のデータセットとして、Livedoor ニュースコーパス<sup>\*12</sup>の一部を採用した。当該コーパスは9つのカテゴリに分類されたニュース記事群からなるラベルを持たないデータセットであるが、本研究では我々の過去の研究で作成した「各カテゴリ毎に100件ずつ記事タイトルを抽出し、所属カテゴリを表すラベルを付与したデータセット」[15][10]を使用した。

#### 4.2 k-NN 分類器による文識別での評価

2つ目の評価手法としては単純な分類器を使用した。k-NN法を採用したシンプルな分類器を用いて文に対する分類問題を解き、その精度を比較するものである。本研究ではデータセットとして筑波大学文単位評価極性タグ付きコーパス(TSUKUBAコーパス)[13]を使用した。当該データセットは楽天トラベルのレビューデータに対し評価極性タグが文単位で付与されたデータセットであり、本研究では収録内容のうちの2674文を使用し、各ラベル毎に

8割を訓練データ、2割をテストデータとして使用するよう分割した。その内訳を表4に示す。

表 4 使用した TSUKUBA コーパスデータのラベル分布

|       | 訓練データ | テストデータ | 合計    |
|-------|-------|--------|-------|
| ラベル P | 1,476 | 369    | 1,845 |
| ラベル K | 662   | 165    | 827   |
| 合計    | 2,138 | 534    | 2,674 |

そして訓練データ・テストデータを各モデルに入力し、得られた埋め込み表現を用いて5-NN分類器の訓練・テストデータに対する正解率の記録・比較を行った。

#### 5. 評価結果

1節において適当な学習済みモデルが存在しないと述べたが、日本語で事前学習されたSentenceBERTが全く公開されていない訳ではない。日鉄ソリューションズ株式会社に所属している園部勲氏が日本語で事前学習されたSentenceBERTを公開している<sup>\*13</sup>。東北大版のBERTをベースとしたSentenceBERTであるものの、事前学習コーパス・ハイパーパラメータ等は公開されていない。本研究ではこのモデルをSBERT-jpと表記する。そして当該モデルに対する評価結果をベースラインとし、他のBERTから構築したSentenceBERTと比較した。

まずクラス内分散とクラス間分散の比を用いた評価結果を表5及び図3に示す。 $A_m$ 及び $B_m$ はそれぞれ、4.1節で述べたクラス内分散の平均、クラス間分散の平均を簡略化した指標であり、評価値が小さいほどより適切な文の埋め込み表現が作成されていることを意味する。

表 5 文書クラスタリングによる評価結果

| モデル                | $A_m$      | $B_m$  | 評価値      |
|--------------------|------------|--------|----------|
| SBERT-jp           | 222,138.34 | 212.94 | 1,043.19 |
| 京大版 SBERT          | 204,566.35 | 253.74 | 806.20   |
| Stockmark 社版 SBERT | 230,969.95 | 103.85 | 2,224.04 |
| SP 版 SBERT         | 310,377.36 | 237.53 | 1,306.70 |
| 東北大版 SBERT         | 179,802.27 | 252.77 | 711.32   |
| NICT 版 SBERT       | 197,759.28 | 268.30 | 737.10   |
| Laboro 社版 SBERT    | 211,038.32 | 222.27 | 949.48   |

<sup>\*12</sup> <https://www.rondhuit.com/download.html#ldcc>

<sup>\*13</sup> <https://qiita.com/sonoisa/items/1df94d0a98cd4f209051>

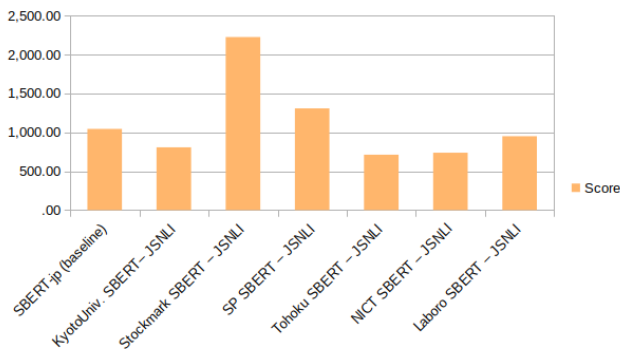


図 3 文書クラスタリングによる評価結果 (Smaller is better)

次に 5-NN 分類器による評価結果を表 6 及び図 4 に示す。

表 6 5-NN 分類器による評価結果 (%)

| モデル                | 評価値   |
|--------------------|-------|
| SBERT-jp           | 94.19 |
| 京大版 SBERT          | 85.96 |
| Stockmark 社版 SBERT | 76.97 |
| SP 版 SBERT         | 81.84 |
| 東北大版 SBERT         | 90.64 |
| NICT 版 SBERT       | 89.33 |
| Laboro 社版 SBERT    | 86.33 |

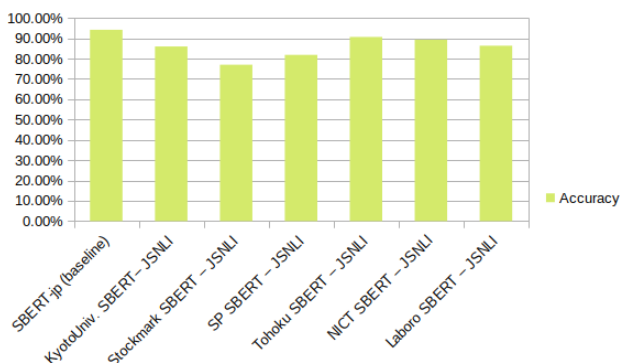


図 4 文書クラスタリングによる評価結果

文書クラスタリングによる評価では、東北大版と NICT 版が同程度の評価値を記録し、京大版がこれに続いている。5-NN 分類器による評価でも同様に東北大版と NICT 版が優秀であり、Laboro 社版と京大版の順位が入れ替わっている。また、文書クラスタリングによる評価ではベースラインとした SBERT-jp を上回る評価値が確認できたが、5-NN 分類器による評価では SBERT-jp より評価値が高いモデルは存在していなかった。

前述した実験結果より、既存の日本語 BERT を用いて SentenceBERT を構築する場合、東北大版 BERT または NICT 版 BERT のいずれかをベースとするのが良い事が示された。

## 6. 考察

### 6.1 SentenceBERT の fine-tuning

SentenceBERT は BERT よりもより適切な文の埋め込み表現を構築できることは明らかである。ただしその埋め込み表現自体は fine-tuning に適したものではない可能性がある。ここでは文書分類のタスクを利用してこの点を確認する。

実験データとしては livedoor ニュースコーパスを用いる。前節ではこのデータの記事タイトルを用いたが、ここでは記事の文書をデータとする。livedoor ニュースコーパスは 7,376 記事のデータからなり、各記事は 9 つのクラスのいずれかに属する。このコーパスをランダムに 10 等分し、そのうちの 2 つを取り出し、それらを training data 及び test data として利用した。そして BERT あるいは SentenceBERT の feature based の利用により文書分類を行った。つまり BERT 部分のパラメータは固定し、その上層の分類層だけを学習する形である。training data を用いて 20 epoch まで学習し、各 epoch 毎に作成できたモデルに対して test data の識別精度を測った。

東北大版 BERT と東北大版 SentenceBERT を比較した結果を図 5 に示す。

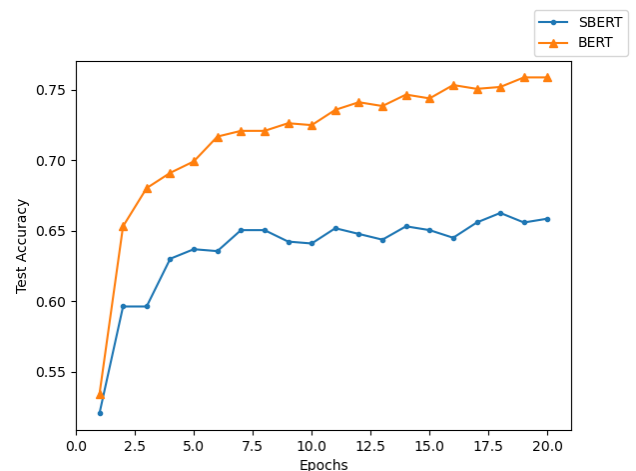


図 5 東北大版 BERT と東北大版 SentenceBERT の feature-based による精度比較

次に NICT 版 BERT と NICT 版 SentenceBERT を比較した結果を図 6 に示す。

図 5 と図 6 をみると学習全体において BERT は明らかに SentenceBERT よりも高い精度を出しており、学習を続けるほど差は大きくなる傾向が見られた。

これは SentenceBERT の出力する文の埋め込み表現は、上層のネットワークに対して適した形になっているとは限らないことを示している。また SentenceBERT は fine-tuning の効果があまり出ないことも示している。どのような場合に SentenceBERT の fine-tuning の効果が出るかを

表 7 SentenceBERT による One-shot 学習の正解率 (%)

| モデル          | 1 回目  | 2 回目  | 3 回目  | 4 回目  | 5 回目  | 平均    |
|--------------|-------|-------|-------|-------|-------|-------|
| 東北大版 SBERT   | 77.49 | 75.05 | 45.78 | 78.61 | 79.55 | 71.30 |
| NICT 版 SBERT | 61.54 | 51.78 | 46.53 | 73.17 | 82.55 | 63.11 |

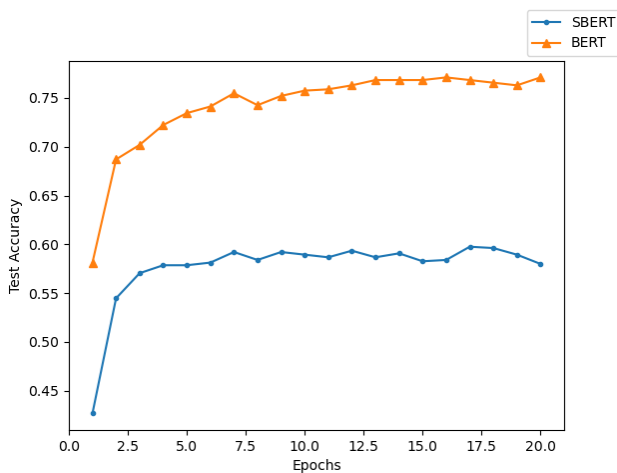


図 6 NICT 版 BERT と NICT 版 SentenceBERT の feature-based による精度比較

今後調べていきたい。

## 6.2 One-Shot 学習への利用

我々は SentenceBERT の一つの応用として文分類の One-Shot 学習を検討している。SentenceBERT がより適切な文の埋め込み表現を構築できるのであれば、文分類を行う際の訓練データでは小規模であっても高い性能が期待できる。ここでは前節で利用した TSUKUBA コーパスを用いて、この点を確認する。前節で利用した TSUKUBA コーパスの訓練データから Positive (ラベル P) の文と Negative (ラベル K) の文を 1 つずつランダムに選択し、それらを SentenceBERT により文の埋め込み表現に変換する。これが訓練データとなる。またテストデータも SentenceBERT により文の埋め込み表現に変換しておく。以上から最近傍法 (1-NN) を利用して、文の識別精度を測る。この実験を東北大版 SentenceBERT と NICT 版 SentenceBERT を用いて、5 回行った。その結果を表 7 に示す。

バラツキはあるものの東北大版 SentenceBERT では 0.7 程度の精度は出ており、訓練データの選択をランダムに行っていることを考慮すると、高い数値だと考えられる。訓練データの選択方法やラベル無しデータの利用を検討することで、更に性能は改善できると考えられるため SentenceBERT の One-Shot 学習への利用を今後検討していきたい。

## 7. おわりに

本論文では我々が構築した 6 つの日本語 SentenceBERT とその評価法、及びその評価結果について紹介した。構

築には京都大学で公開されている JSNLI と 6 つの日本語 BERT を用いて行った。また SentenceBERT モデルの評価方法として、クラス内分散とクラス間分散の比を測定する方法、及び k-NN 法による単純な分類器での文の分類問題に対する精度を用いる方法を提案した。提案した評価法による結果では東北大版 BERT と NICT 版 BERT から構築した SentenceBERT が同程度の高い性能を示した。今後は考察の節で述べた SentenceBERT の fine-tuning の効果と One-Shot 学習への利用可能性について調べていきたい。

## 謝辞

本研究は JSPS 科研費 JP19K12093 および 2021 年度国立情報学研究所公募型共同研究 (2021-FC05) の助成を受けています。

## 参考文献

- [1] Bowman, S. R., Angeli, G., Potts, C. and Manning, C. D.: A large annotated corpus for learning natural language inference, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics (2015).
- [2] Cer, D., Yang, Y., yi Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strophe, B. and Kurzweil, R.: Universal Sentence Encoder (2018).
- [3] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- [4] Gencoglu, O.: Sentence Transformers and Bayesian Optimization for Adverse Drug Effect Detection from Twitter, *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pp. 161–164 (2020).
- [5] Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R. and Fidler, S.: Skip-Thought Vectors, *arXiv preprint arXiv:1506.06726* (2015).
- [6] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *Proceedings of the 31st International Conference on Machine Learning (Xing, E. P. and Jebara, T., eds.)*, Proceedings of Machine Learning Research, Vol. 32, No. 2, Beijing, China, PMLR, pp. 1188–1196 (online), available from (<http://proceedings.mlr.press/v32/le14.html>) (2014).
- [7] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space (2013).
- [8] Ndukwe, I. G., Amadi, C. E., Nkomo, L. M. and Daniel, B. K.: Automatic Grading System Using SentenceBERT Network, *International Conference on Artificial*

- Intelligence in Education*, Springer, pp. 224–227 (2020).
- [9] Reimers, N. and Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, (online), available from <https://arxiv.org/abs/1908.10084> (2019).
- [10] Shibayama, N., Cao, R., Bai, J., Ma, W. and Shinnou, H.: Evaluation of Pretrained BERT Model by Using Sentence Clustering, *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, Hanoi, Vietnam, Association for Computational Linguistics, pp. 279–285 (online), available from <https://www.aclweb.org/anthology/2020.paclic-1.32> (2020).
- [11] Shirafuji, D., Kameya, H., Rzepka, R. and Araki, K.: 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* (2020).
- [12] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. et al.: HuggingFace’s Transformers: State-of-the-art natural language processing, *arXiv preprint arXiv:1910.03771* (2019).
- [13] 楽天グループ株式会社: 筑波大学文単位評価極性タグ付きコーパス (TSUKUBA コーパス), *text(tsv)* (2014).
- [14] 吉越卓見, 河原大輔, 黒橋禎夫ほか: 機械翻訳を用いた自然言語推論データセットの多言語化, 研究報告自然言語処理 (NL), Vol. 2020, No. 6, pp. 1–8 (2020).
- [15] 芝山直希, 曹鋭, 白静, 馬ブン, 新納浩幸: 文のクラスタリングを用いた BERT 事前学習モデルの評価, 言語処理学会第 26 回年次大会, pp. 1233–1236 (2020).