文のクラスタリングを用いた BERT 事前学習モデルの評価

芝山直希　曹鋭　白静　馬ブン　新納浩幸
茨城大学大学院理工学研究科情報工学専攻

{19nm714t, 18nd305g, 19nd301r, 19nd302h, hiroyuki.shinnou.0828}@vc.ibaraki.ac.jp

1 はじめに

本論文では日本語 BERT モデルを評価するために、文のクラスタリングを用いることを提案する。また提案手法を利用して現在公開されている 4 つの日本語 BERT モデルを評価した。

BERT [1] は高性能な事前学習モデルであり、BERT を利用することで様々な自然言語処理タスクの性能が向上している。BERT のような事前学習モデルを評価するには、一般に、タスクベースのアプローチが取られる。英語であれば GLUE [3] のようなタスクベースの評価用データセットが用意されているために、モデルの優劣を評価するのは容易である。ただしタスクベースで評価する際に、事前学習モデルを fine-tuning するために、fine-tuning のためのメタパラメータの設定により評価値が変化する可能性がある。そのためタスクベースで fine-tuning を行うと、公正な評価にはなっていない可能性がある。また英語以外の言語に対する事前学習モデルに対しては、通常、共通に利用できる評価用データセットが存在しないために、複数の事前学習モデルの優劣を競わせることが難しい。

本論文では日本語の BERT モデルを対象に、入力文に対して出力される CLS トークンの埋め込み表現を評価することでモデルの評価を行う。CLS トークンの埋め込み表現は入力文に対する埋め込み表現と見なすことができるので、この埋め込み表現自体を評価することでモデルの評価が行える。また埋め込み表現自体を評価すれば fine-tuning の問題は避けられる。ただし文の埋め込み表現をどう評価するかも難しい問題である。本論文では文の埋め込み表現を評価するために、クラスタリングを利用する。ジャンル別の文のセットを用意しておき、各文から BERT モデルを利用して、その埋め込み表現を取り出す。それら埋め込み表現をクラスタリングし、クラスタリングの評価値を利用してモデルを評価する。

実験では日本語の BERT モデルとして、京都大学の黒橋・河原研究室で公開した BERT モデル（「京大版」と略す）[*1]、森長氏が公開している BERT モデル（「MeCab 版」と略す）[*2]、Yohei Kikuta 氏が公開している BERT モデル（「SP 版」と略す）[*3] 及び東北大学の乾・鈴木研究室が公開している BERT モデル（「東北大版」と略す）[*4] の 4 つを対象とする。またジャンル別の文のセットとしては、livedoor ニュースコーパスの各カテゴリの記事からタイトルを取り出して利用する。実験の結果 MeCab 版が最もよい評価値を出した。

2 関連研究

BERT などの事前学習モデルどうしを比較して評価するには、タスクベースの評価法を用いるのが一般的である。あるタスクに対して事前学習モデル A を用いてタスクの精度を測る。次に先の事前学習モデル A を事前学習モデル B に置き換えて同様に精度を測る。得られた 2 つの精度の比較によりモデル A と B を評価するというものである。これは単純な評価法であるが、評価用のデータを必要とする。英語に対しては GLUE が存在するが、他の言語に対しては標準的なデータが存在しないために、独自に評価用データを準備するしかない。

複数の日本語の BERT モデルを比較評価した研究としては、我々のグループが Amazon dataset の文書分類のタスクを用いて、京大版、MeCab 版、SP 版の 3 つを評価した [5]。ただし BERT は文に対するモデルであり、文書を対象とした文書分類をどのように行うかは、まだ、確定した手法は存在しない。そのためにタスクとして文書分類を用いるのは適切かどうか疑問

[*1] http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT 日本語 Pretrained モデル
[*2] https://qiita.com/mkt3/items/3c1278339ff1bcc0187f
[*3] https://github.com/yoheikikuta/bert-japanese
[*4] https://github.com/cl-tohoku/bert-japanese

である。本論文で対象とする BERT への入力は文であり、入力文の埋め込み表現と考えられる BERT の出力する CLS トークンの埋め込み表現を利用してモデルの評価を行う。

埋め込み表現を評価するにもタスクベースで行うしかないが、word2vec [2] などにより得られる単語の埋め込み表現に関しては、その埋め込み表現が単語の意味を表現しているという観点から、人間が手作業で与えた単語間の類似度と、埋め込み表現から得られる単語間の類似度との相関を測ることで埋め込み表現の評価も行われている [4]。

3 BERT の評価

第 2 節で述べたように、評価対象となる事前学習済み BERT モデルが入力文に対して出力する CLS トークンの埋め込み表現を評価する場合、通常タスクベースになる。一方、同節で述べたように word2vec などの場合、ある埋め込み表現が単語の意味を表現しているという観点を有する。この観点をクラスタリングに適用すると、「あるクラスタはそのクラスタに属する埋め込み表現の集合で表現できる」と述べることができる。

本論文ではこれを利用し、文のクラスタリングを用いた事前学習済み BERT モデルの評価を行う。

3.1 評価方法

本論文では BERT モデル m が出力する特徴量ベクトルに対して、以下のような手順で評価する。なお、モデル m に入力する文には正解クラスのラベルが設定されている必要がある。

1. BERT モデル m の各文の出力のうち、 CLS トークンのベクトルを抽出し、文のベクトルとする。
2. 文ベクトルが属する正解クラスを確認し、モデル m の各クラスの重心 $g_i^{(m)}$ を導出する。
3. 各クラスのクラス内分散 A_m を以下から計算する[*5]。

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

ここで $\sigma_i^2 = \sum_{j \in C_i} ||g_i^{(m)} - x_{i,j}||^2$ であり N はクラス数を表す。

4. モデル m の全クラスの重心の平均 $g^{(m)}$ を導出し、クラス外分散

$$B_m = \sum_{i=1}^{N} ||g^{(m)} - g_i^{(m)}||^2 (N = \text{クラスの個数}) \tag{2}$$

を計算し、B_m を導出する。[*6]

5. 分離度 $M_m = \frac{A_m}{B_m}$ を計算し、これをモデル m の評価値とする。この値は適切にクラスタリングできている場合により小さくなる。

4 実験

4.1 実験設定

本実験では livedoor ニュースコーパス[*7]を使用した。これは livedoor ニュースの記事のうち、以下の 9 カテゴリに属するものを収集したデータセットである。

- トピックニュース
- Sports Watch
- IT ライフハック
- 家電チャンネル
- MOVIE ENTER
- 独女通信
- エスマックス
- livedoor HOMME
- Peachy

本実験では全 9 カテゴリのそれぞれに対し、ニュース記事 100 件を選択して利用した。評価対象とした事前学習済み BERT モデルは京都大学の黒橋・河原研究室で公開した BERT モデル（「京大版」と略す）、森長氏が公開している BERT モデル（「MeCab 版」と略す）、Yohei Kikuta 氏が公開している BERT モデル（「SP 版」と略す）及び東北大学の乾・鈴木研究室が公開している BERT モデル（「東北大版」と略す）の 4 つである。表 1 に各モデルの特徴となる tokenizer と訓練コーパスを示す。

4.2 実験結果

まず、選択された記事から記事タイトルのみを抽出した。その後、これにより得られた全 900 タイトルを 1 タイトル 1 文とみなして比較対象とする BERT モデルに入力し、出力を得た。獲得した出力から [CLS] の特徴量ベクトルをそのタイトルの埋め込み表現とし、抽

[*5] 本論文では簡略化のために偏差の二乗を分散とみなす。そのため σ_i^2/N を計算することで本来のクラス内分散を導出できる。

[*6] A_m の場合と同様に、全クラスの重心の偏差の二乗を分散とみなしている。

[*7] http://www.rondhuit.com/download.html\#ldcc

表 1　各 BERT モデルの評価値

モデル	tokenizer	訓練コーパス
京大版	Juman++	Wikipedia
MeCab 版	MeCab + NEologd	ビジネスニュース記事
SP 版	SentencePiece	Wikipedia
東北大版	MeCab + NEologd	Wikipedia

出した。

各 BERT モデルで得られた各タイトルの埋め込み表現に対し、カテゴリ 1 つをクラス 1 つとみなした上で 3.1 節にて述べた評価方法を用いて入力文のクラスタリングを行い、各モデルの評価値を計算した。

各モデルの評価値及び A_m、B_m の値は表 2 のようになった。

表 2　各 BERT モデルの評価値

モデル	A_m	B_m	評価値
京大版	240164.31	337.84	710.88
MeCab 版	74470.61	162.53	**458.19**
SP 版	65220.69	97.50	668.92
東北大版	48720.21	61.49	792.34

5　考察

5.1　評価値の傾向

表 2 で示している評価値を単純比較すれば、モデルの性能は良い順に以下となる。

MeCab 版 > SP 版 > 京大版 > 東北大版

このような評価結果となった原因を A_m、B_m の性質を用いて考察する。3.1 節の脚注で述べたように、A_m、B_m は評価値計算の簡略化により厳密には異なるものの、それぞれクラス内分散、クラス外分散とみなすことができる。よって、A_m、B_m の比較は、各モデルが「同クラスの文ベクトルが類似するよう出力したか」「異なるクラスのベクトルが類似しないよう出力したか」という 2 点で評価し、それぞれで比較することと同義だとみなすことができる。本節ではまず A_m の傾向を確認し、その後 B_m の傾向からモデルの比較を行っていく。

表 2 によれば、A_m は小さい順に以下となる。

東北大版 < SP 版 < MeCab 版 < 京大版

同クラスの文ベクトルが最も「まとまっている」のは東北版のモデルだった。その後に SP 版、MeCab 版、京大版が続く。一方 B_m は、大きい順に以下となった。

京大版 > MeCab 版 > SP 版 > 東北大版

異なるクラスの文ベクトルが「離れている」順では京大版をトップに MeCab 版、SP 版、東北大版 が続いている。

以上の A_m、B_m の比較結果及び表 2 から、各モデルの大まかな傾向を述べることができる。A_m こそ 3 位ではあるものの、B_m において SP 版のモデルより 50 以上大きい MeCab 版が最優となった。それに続いたのが MeCab 版以下の A_m を記録したものの、異なるクラスの文ベクトルの距離が MeCab 版より近かった SP 版である。評価値が最下位となった東北大版のモデルは、同クラスの文ベクトルは最もまとまっているがクラス間の距離が最も近いという傾向だった。これは他のモデルに比べ全ベクトルのばらつきが小さいことを示している。

5.2　分類による評価

本研究で生成された各埋込み表現に対し、分類器の訓練及び 9 クラス分類の正解率の測定を 5 回行い、比較した。分類器の作成・訓練にあたり、フレームワークは pytorch を使用した。また、各カテゴリの末尾 20% をテストデータとし、残った 720 文のベクトルを分類器の訓練に使用した。分類器のモデル構成を図 1 に示す。分類器の optimizer は Adam を使用した。また、分類器の各中間層は 200 次元であり、訓練は 20epoch 行った。

この検証実験により得られた正解率の最小値・最大値及び平均値を表 3 に示す。

この検証実験結果を平均値で比較すると、僅差で以下となった。

京大版 > SP 版 > MeCab 版 > 東北大版

この検証結果は東北大版以外の BERT モデルの順位が表 2 から得られる評価値の順位と異なる結果である。

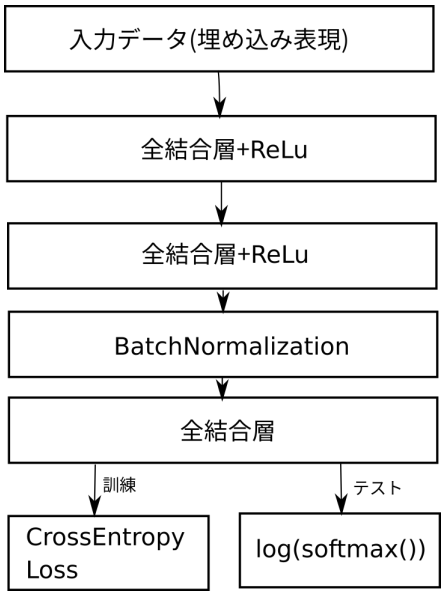

図 1　分類器の構成

表 3　分類正解率の概要

モデル	最小値	最大値	平均値
京大版	**68.89%**	**71.11%**	**70.00%**
MeCab 版	67.78%	70.00%	69.11%
SP 版	**68.89%**	70.56%	69.78%
東北大版	66.11%	70.00%	68.67%

よって、この検証実験では明確な性能差のあるモデルの比較は、3.1 節で述べた評価法で問題なく評価できることが確認された。一方、表 3 で示された結果は東北大版以外の評価が適切であると断言できるものではなかった。

この結果の遠因としてクラス分類による検証実験でのテストデータ抽出が考えられる。本節で述べたように各クラスの末尾から 20% をテストデータとしているが、各クラスの末尾 20% の傾向と各クラス全体の傾向が一致しているとは限らない。「複数回測定して BERT モデルの比較をする」という目的を考慮すると、各クラスの 20% をランダムに選択しテストデータとするのが適切である。

6　おわりに

本研究では正解クラスが存在する文集合及びそれらを入力した BERT の出力を利用し、日本語で事前訓練された BERT モデルの評価を行った。その結果、優秀である順に以下となった。

MeCab 版 > SP 版 > 京大版 > 東北大版

また、クラス分類の正解率による各モデルの再評価を行った。しかし、3.1 節で述べた評価法による評価結果と正解率による評価結果は、東北大版の順位は同一であったものの、他のモデルの順位が異なる結果となった。この結果から 3.1 節で述べた評価方法はわずかな性能差のあるモデルを適切に評価できるとは言えないが、一定以上の性能差があるモデルを適切に評価できることが判明した。

今後取り組むべき事項としてテストデータをクラス分布が均衡になるようにランダムに選択し、検証実験を再度行うのがまず挙げられる。その後 3.1 節の評価方法での結果と再検証実験での結果を比較し、本研究での評価手法及びそれを用いた評価自体に対する評価を確固たるものにしていきたい。

参考文献

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

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

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

[4] 小町守, 堺澤勇也. 日本語動詞・形容詞類似度データセットの構築. 言語処理学会第 22 回年次大会, pages 258–261, 2016.

[5] 芝山直希, 曹鋭, 白静, 馬ブン, 新納浩幸. 日本語 pretrained bert モデルの比較. 21, 2019.