Vol.2020-AAC-12 No.4
2020/3/13

事前学習済み BERT の単語埋め込みベクトルによる同形異音語の読み誤りの改善

佐藤文一[†1]　喜連川優[†2]

概要：重度視覚障害者は，パソコンで画面読み上げソフトを使用して，漢字交じりの文書を，音声で聞いている．しかし，「表に出る」を「ひょうにでる」と読み上げられると理解が困難になるなどのことから，正しい読み上げが必要になる．汎用言語モデルの BERT の日本語事前学習されたデータが公開されており，大量の文で学習しているため，文脈による単語埋め込みベクトルが異なることが期待される．同形異音語（読み方が複数ある単語）として「国立」「表」「大分」に対して，日本語書き言葉均衡コーパスから文章を選び出し，学習・テストで 2 クラス分類の評価を行った．単語埋め込みベクトルを使用した結果の「コサイン類似度」「SVM」による分類の適合率・再現率・F1 値が改善した．

キーワード：視覚障害者，情報障害，BERT，単語埋め込みベクトル

Improvement of Reading of Heteronym by a Word Embedded Vector on Pre-trained BERT

FUMIKAZU SATO[†1]　MASARU KITSUREGAWA[†2]

１．はじめに

「視覚障害者等の読書環境の整備の推進に関する法律」（通称 : 読書バリアフリー法）が 2019 年 6 月に公布・施行され，視覚障害者の読書環境の整備が行われ始めている．

「障害者の学習に対しては，2008 年に障害のある児童及び生徒のための教科用特定図書等の普及の促進等に関する法律」（通称 : 教科書バリアフリー法）が成立し，たとえば視覚障害の小中学校の生徒に対しては拡大・点字教科書の提供が制度化された．障害のある人の情報アクセシビリティを向上するための施策が発表され，行われてきている．しかしながら，視覚障害者の情報障害に対しては，まだ克服すべき課題が多い．

その一つとして「読み上げ」の問題がある．重度視覚障害者は，パソコンで画面読み上げソフトを使用して，漢字交じりの文書を, 音声で聞いている. たとえば,「表に出る」「真理の追究」「大分への出張」のそれぞれの最初の漢字の読みが,「ひょう」「まり」「だいぶ」と読み上げられた場合は，文の理解が困難になる．従って正しく読み上げられることが必要であり，正しく読み上げるためには，文脈を考慮して読みを決定しなければならない．

汎用言語モデルの BERT[1]が 2018 年に発表され, 大量の日本語文章すなわち，大量の文脈を使って学習した「事前学習済み BERT」[2]が公開されている．この学習済み BERT を使用して, 単語埋め込みベクトルにより, 同形異音後（複数の読みがある単語）が区別できるかを調査した．評価に使用する文は，現代日本語書き言葉均衡コーパス]3]を使用した．

本稿の構成は，次の通りである．第 2 節で，本論で使用している BERT 等の要素技術を関連研究として概説する．第 3 節では，予備調査として，まずは少ないサンプルで，単語埋め込みベクトルによって意味の異なる単語が判別できるかの確認を行う．第 4 節で，日本語書き言葉均衡コーパスから文章を選び出し，同形異音語に対して学習・テストで 2 クラス分類の評価を行う．第 5 節でまとめと今後の方向性について述べる．

２．既存研究

自然言語処理の分野では，単語をベクトルで表す研究は盛んに行われてきており，特に 2013 年に Mikolov らが考案した Word2Vec[4] により，語彙数より少ない次元 Embedding 数のベクトルを学習から得る方法が提案された. 周辺の単語から対象単語を推測していく CBoW(Continuous Bag-of-Words)モデル，逆に単語からその周辺単語を予測する Skip-gram(Continuous Skip-gram)モデルである. これにより，似た文脈で出てくる単語同士の類似性をとらえることができるようになった.

2018 年 10 月には, Google の Jacob Devlin らにより, BERT (Bidirectional Encoder Representations from Transformers) が発表され，双方向 Transformer により，大規模コーパスから言語モデルを事前学習することにより，自然言語処理の 11 のタスク(GLUE では 8 タスク)で，SOTA(State of the Art)を達成した．教師ラベルの付いていない文章を使うことによ

†1 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, the University of Tokyo
†2 国立情報学研究所，東京大学生産技術研究所
National Institute of Informatics/Institute of Industrial Science, the University of Tokyo

り，大量のデータで学習を行っている．事前学習の手法として，一つは，Masked LM という手法で，文章の 15%をMASK トークンに置き換え，そのうちの 80%をマスクし，この MASK された単語を予測することにより，前後の文脈を考慮する手法である．二つ目は，Next Sentence Prediction で，2 つの文章が与えられたときに，50%の確率で二つ目の文章を隣接していない別の文章に置き換え，これらが隣接文章かを判定する．

京都大学黒橋・河原研究室が，BERT の日本語に対応した事前学習モデルを公開している．この事前学習モデルの入力となるテキストは，日本語 Wikipedia 全部（約 1,800 万文）で，語彙数は（サブワードも含む）は 32,000 である．

国立国語研究所コーパス開発センターでは，現代日本語の書き言葉の全体像を把握するために,『現代日本語書き言葉均衡コーパス』（BCCWJ）というコーパスを作成している．書籍全般，雑誌全般，新聞，白書，ブログ，ネット掲示板，教科書，法律などのジャンルから，無作為にサンプルを抽出し，1 億 430 万語のデータを格納している．また，すべてのサンプルは長短ふたつの言語単位を用いて形態素解析されている．

表 1 単語の cosine 類似度

コサイン類似度	単語1	文	単語の位置	単語2	文	単語の位置
0.6860	表(おもて)	文A1	位置4	表(おもて)	文A1	位置15
0.5609	表(おもて)	文A1	位置4	裏	文A1	位置6
0.5552	裏(うら)	文A1	位置6	表(おもて)	文A1	位置15
0.5425	表(おもて)	文A1	位置15	表(ひょう)	文B2	位置10
0.4766	表(おもて)	文A1	位置4	表(ひょう)	文B2	位置10
0.4203	簡単(かんたん)	文B2	位置4	表(ひょう)	文B2	位置10
0.3953	裏(うら)	文A1	位置6	表(ひょう)	文B2	位置10
0.3114	表(おもて)	文A1	位置15	簡単	文B2	位置4
0.2099	裏	文A1	位置6	簡単	文B2	位置4
0.1677	表(おもて)	文A1	位置4	簡単	文B2	位置4
0.6743	肩	文B2	位置6	首(体)	文B2	位置8
0.5893	首(解雇)	文B1	位置6	解雇	文B1	位置11
0.3830	解雇	文B1	位置11	仕事	文B1	位置14
0.2783	首(解雇)	文B1	位置6	仕事	文B1	位置14
0.2775	首(解雇)	文B1	位置6	首(体)	文B2	位置8
0.2635	仕事	文B1	位置14	肩	文B2	位置6
0.2520	仕事	文B1	位置14	首(体)	文B2	位置8
0.2112	解雇	文B1	位置11	首(体)	文B2	位置8
0.1994	首(解雇)	文B1	位置6	肩	文B2	位置6
0.1697	解雇	文B1	位置11	肩	文B2	位置6

＊なお、表 1 の単語の位置は、BERT により先頭に CLS が加わるため、1 大きい値になっている。

３．予備調査

3.1 単語埋め込みベクトルの文脈による判別のトライアル

大規模な現代日本語書き言葉均衡コーパスで検証する前に，BERT で作成した単語埋め込みベクトルで，単語の使い方が文脈により区別できているかの予備確認を行った．なお，英文では，bank(銀行)と bank(土手)に対して，分類できることが知られている．

方法としては,まず形態素解析エンジン JANOME[5]を使用して単語に分割し，前節で概説した日本語事前学習済み BERT から，単語の埋め込みベクトルを算出し，単語のベクトルの内積によるコサイン類似度を算出する．

以下は確認のための文と，その単語埋め込みベクトルのコサイン類似度の計算結果である．

文 A1 「紙の表と裏は，つるつるする方が表である.」
文 A2 「内容を簡単に比較できるように表を作成した.」

文 B1 「彼は会社を首になり，つまり解雇になり仕事を探している.」
文 B2 「疲れたので，肩と首をマッサージした.」

「表（おもて)」の対義語の「裏」の方が「表（ひょう)」よりコサイン類似度が近いという結果が得られた．また，「首（解雇)」と「解雇」の間のコサイン類似度,「首（体)」と「肩」のコサイン類似度も近いという結果が得られた．

以上により，コサイン類似度により,「表（おもて)」と「表（ひょう)」が区別できそうという結果が得られた．同様に解雇の意味の「首」と体の「首」も区別できそうという結果が得られた．予備調査で，区別できそうな結果が得られたので,サンプル数を増やして確認を行うことにした．

3.2 現代日本語書き言葉均衡コーパスからの学習・評価のためのデータ数の確認

現代日本語書き言葉均衡コーパスの中から，同形異音語として,「国立」「一言」「表」「大分」「一行」「最中」「角」

表 2 BCCWJ の同形異音語のふりがなの出現数

						BCCWJ					分類時	
漢字	ふりがな0	ふりがな1	ふりがな2	ふりがな3	ふりがな4	ふりがな0の数	ふりがな1の数	ふりがな2の数	ふりがな3の数	ふりがな4の数	ラベル0 ふりがな0の数	ラベル1 ふりがな1の数
国立	こくりつ	くにたち				3583	61				3540	61
一言	ひとこと	いちげん	いちごん	一言		3362	119	11	1		3352	116
表	ひょう	おもて	表	あらわし		22841	375	33	2		22455	367
大分	おおいた	だいぶ	大分	だいぶん		1618	270	1	1		1601	268
一行	いっこう	一行				1531	4				1364	2
最中	さいちゅう	さなか	最中			1666	1	1			1458	1
角	かく	かど	すみ	つの	角	2020	1411	28	6	1	1992	1383
人気	にんき	ひとけ	ひとげ			9039	2	1			8515	2
一目	ひとめ	いちもく				727	477				694	219
上方	かみがた	じょうほう				614	202				613	201
上手	じょうず	かみて	うま	うわて	じょうて	3147	354	38	5	2	3053	339
人事	じんじ	ひとごと				3222	1				3201	1

「人気」「一目」「上方」「上手」「人事」の 12 単語を選び，その出現数を調べた．

表 2 は，上記の単語の読みの出現数と，BERT の単語埋め込みベクトルを算出した後の数を示している．数が若干減っている理由の一つは，BERT により sub-word に分解され，BERT token にされるが，事前学習済み BERT では，その数が最大 126 個という制限があるためである．評価は単語の読みの種類を 2 個のみ使用し，2 クラス分類で行うことにした．このため，たとえば国立の読み「こくりつ」と「くにたち」を含む文のそれぞれが，分類を行うためには，ある程度の数が必要である．読みの一方の数が極端に少ない，「一行」「最中」「人気」」人事」は除外することにした．

以上の結果より，現代日本語書き言葉均衡コーパスを使用することにより，同形異音語の分類の学習・評価ができると判断した．

3.3 現代日本語書き言葉均衡コーパスからの正解ラベルの作成

現代日本語書き言葉均衡コーパスは，分かち書きされており，ふりがなもふられているが，誤りも含まれていることが判明した．従って，このままでは分類の学習・評価には使用できない．

正解ラベルを作成するために，ふりがなが確定している複合語，たとえば国立大学，国立市，時刻表等は，プログラムで正解ラベルを割り当て，残りに対しては，人手で正解ラベルを割り当てた．「国立」，「大分」は，ほぼ全数に対して正解ラベルを割り当てた．「表」は，8 割に対して割り当てた．人手で行うため，残りの単語は正解ラベルを作成するのを断念し，「国立」「表」「大分」に対して評価を行うことにした．

4．同形異音語の学習・評価

4.1 形態素解析エンジンの JANOME による評価

JANOME を使用すると，各単語に対して，

表層形¥t 品詞,品詞細分類 1,品詞細分類 2,品詞細分類 3,活用形,活用型,原形,読み,発音

の「読み」より，ふりがなを取得できる．

今回使用学習済み BERT は，sub-word に展開されているので，JANOME の形態素で得られた単語に対して，同様に sub-word への展開を行っている．たとえば「表参道」は「表」と「参道」に展開している．

表 3 は，複数の分類結果の適合率（precision），再現率（recall），F1 値をまとめたものである．

適合率は，どれくらい正しいかを示す指標で，陽性と予測した中で正解した割合である．再現率は，正解をどれくらい漏れなく予測できたかの指標で，実際に陽性の中での正解の割合である．F1 値は適合率と再現率の調和平均である．

表 3 の「JANOME」の列が JANOME を使用した場合の，適合率，再現率，F1 値の結果である．「国立」「表」「大分」のラベル 1 の recall は，それぞれ，0.129，0.261，0.329 であった．大分（おおいた）は名刺であり，大分（だいぶ）は副詞であるので，品詞分類も行える形態素解析の結果は，予想より低い値であった．

なお，表 3 の「BCCWJ」の列は，現代日本語書き言葉均衡コーパスの元のふりがな 0（ラベル 0）とふりがな 1（ラベル 1）に対する，適合率，再現率，F1 値を算出した結果である．「国立」「表」「大分」のラベル 1 の recall は，それぞれ，0.645，0.191，0.347 であった．

この結果からもわかるように，正解ラベルを作成するための作業が必要であった．

4.2 単語埋め込みベクトルのコサイン類似度による評価

評価は 2 つの方法で行った．方法 1 では，学習データのラベル 0 とラベル 1 のそれぞれの単語埋め込みベクトルの平均を求める．テストデータの単語に対して，上記平均ベクトルとのコサイン類似度を計算し，類似度の大きい方のラベルを正解として予測する．方法 2 は，方法 1 の二つの平均ベクトルに対して，学習データに対してコサイン類似度でラベルを予測し，正解ラベルが 0 で予測が 0，正解ラベルが 0 で予測が 1，正解ラベルが 1 で予測が 0，正解ラベルが 1 で予測が 1 の 4 つの集合に分け，それぞれの平均ベクトルを計算する．

次にテストデータの単語に対して，上記の 4 つの平均ベクトルとのコサイン類似度を計算し，類似度の大きい方のラベルを正解として予測する．データを学習 50%,テスト

表 3　JANOME 同形異音語の分類結果

漢字		BCCWJ	JANOME	AVERAGE 類似度(2nd)	SVM (全平均)		BCCWJ	JANOME	AVERAGE 類似度(2nd)	SVM (全平均)	ラベル0の サンプル数	ラベル1の サンプル数
国立	ラベル0(こくりつ)のprecision	0.991	0.977	0.994	0.994	ラベル1(くにたち)のpr	0.984	0.800	0.915	0.971	3507	93
	recall	1.000	0.999	0.998	0.999	recall	0.645	0.129	0.796	0.786		
	f1	0.995	0.988	0.996	0.997	f1	0.779	0.222	0.851	0.866		
表	ラベル0(ひょう)のprecision	0.922	0.924	0.996	0.998	ラベル1(おもて)のpre	0.981	0.333	0.801	0.985	17769	1869
	recall	1.000	0.945	0.974	0.999	recall	0.191	0.261	0.964	0.974		
	f1	0.959	0.934	0.985	0.998	f1	0.320	0.293	0.875	0.980		
大分	ラベル0(おおいた)のprecision	0.685	0.680	0.998	0.989	ラベル1(だいぶ)のpre	0.985	0.992	0.905	0.981	1084	760
	recall	0.996	0.998	0.926	0.986	recall	0.347	0.329	0.997	0.984		
	f1	0.812	0.809	0.961	0.988	f1	0.514	0.494	0.949	0.982		

50%に分割して評価を行った．表 3 の「AVERAGE 類似度(2nd」の列は，方法 2 の適合率，再現率，F1 値の結果を示す．なお，方法 2 の方が結果が良かったので，方法 2 のみを示している.「国立」「表」「大分」のラベル 1 の recall は，それぞれ，0.796，0.964，0.997 であった．

4.3 単語埋め込みベクトルの scikit-learn のサポートベクターマシン(SVM)による評価

機械学習ライブラリーscikit-learn[7]の GridSearchCV を使い，ハイパーパラメータの最適化を行い，サポートベクターマシン(SVM)でクラス分類を行った．GridSearchCV を使うことにより，指定したパラメタの全部の組合せの中から Cross validation で最適なパラメタを決定することができる．GridSearchCV を学習データに対して行い，この時の交差検定は，デフォルトの 3 で行った．カーネルの種類としては rbf（ガウスカーネル）と linear カーネルで行ったが，最適なカーネルは rbf が選ばれた．またデータを学習 50%とテスト 50%に分割した．

表 3 の「SVM(全平均)」の列は，適合率，再現率，F1 値の結果を示す.「国立」「表」「大分」のラベル 1 の recall は，それぞれ，0.786，0.974，0.984 であった．なお，表の（全平均）は，学習データとテストデータを入れ替えた平均である．

4.4 mecab-ipadic-NEologd による評価

以上 4.1 から 4.3 は，形態素解析エンジンとしては JANOME を使用した.「表参道」のような複合語は「表」と「参道」に分解されふりがなに誤りが発生する場合がある．形態素解析エンジン MeCab と固有表現が多数登録されている mecab-ipadic-NEologd[6]（以下，NEOLOGD と呼ぶ）を使用することにより,「表参道」の固有表現でふりがなが登録されており,「表」と「参道」に分割されない．これを利用して, 読みが確定している固有表現（複合語）は, クラス分類の対象から除外することにした.「国立」「表」「大分」のラベル 0 とラベル 1 のサンプル数は, それぞれ, (3507, 93)->(1118, 38) , (17769, 1869)->(15293, 1537) , (1084, 760)->(447, 757) であった．

分類すべきサンプル数が 1/3 に削減できている．

表 4 は，複数の分類結果の適合率（precision），再現率（recall），F1 値をまとめたものである．

以下 4.1-4.3 説の結果と，今回の NEOLOGD でサンプル数を削減した後の結果を比較する．

NEOLOGD も JANOME と同様に単語の読みが得られる．表 4 の「NEOLOGD」の列より,「国立」のラベル 1「くにたち」は,「こくりつ」に対して頻度が大幅に少ないため, NEOLOGD では意図的かどうかは不明であるが，全て「こくりつ」としている．

「表」「大分」のラベル 1 の recall は，それぞれ，0.261->0.107，0.329->0.309 という結果になった．

現代日本語書き言葉均衡コーパスの元の読みに対して,「国立」「表」「大分」のラベル 1 の recall は，それぞれ, 0.645-> 0.5　0.191->0.193　0.347->0.349 となり，ほぼ同じ結果であった．

「AVERAGE 類似度（2nd)」については,「国立」「表」「大分」のラベル 1 の recall は，それぞれ, 0.796->0.824　0.964->0.949 0.997->0.987 となり，ほぼ同じ結果であった．

「SVM（全平均)」については,「国立」「表」「大分」のラベル 1 の recall は，0.786->0.696，0.974->0.975，0.984->0.989 となり，ほぼ同じ結果であった．

5. おわりに

事前学習済み BERT の単語埋め込みベクトルにより，同形異音語の読みをクラス分類で改善できることを示した．現代日本語書き言葉均衡コーパスの元の読みを base line とすると,「国立」「表」「大分」のラベル 1 の recall は，それぞれ，0.645->0.824，0.191->0.949，0.347->0.987 と大幅に改善している．

計算コストを考えると，下記のステップで行うことを提案する．

1. 該当単語が mecab-ipadic-NEologd の固有表現に含まれているかを調べる．
2. 含まれていない時は，単語埋め込みベクトルを算出する．
3. 4.2 節の方法である，事前に算出している該当単語の各

表 4　NEOLOGDE 同形異音語の分類結果

漢字		BCCWJ	NEOLOGD	AVERAGE 類似度(2nd)	SVM (全平均)		BCCWJ	NEOLOGD	AVERAGE 類似度(2nd)	SVM (全平均)	ラベル0のサンプル数	ラベル1のサンプル数
国立	ラベル0(こくりつ)のprecision	0.983	0.967	0.994	0.990	ラベル1(くにたち)のpr	0.950	-	0.412	0.899	1118	38
	recall	0.999	0.988	0.964	0.997	recall	0.500	-	0.824	0.696		
	f1	0.991	0.977	0.979	0.994	f1	0.655	-	0.549	0.784		
表	ラベル0(ひょう)のprecision	0.925	0.912	0.995	0.997	ラベル1(おもて)のpre	0.977	0.130	0.815	0.984	15293	1537
	recall	1.000	0.928	0.978	0.999	recall	0.193	0.107	0.949	0.975		
	f1	0.961	0.920	0.986	0.998	f1	0.322	0.117	0.877	0.979		
大分	ラベル0(おおいた)のprecision	0.474	0.459	0.976	0.980	ラベル1(だいぶ)のpre	0.992	0.987	0.954	0.982	447	757
	recall	0.996	0.993	0.919	0.969	recall	0.349	0.309	0.987	0.989		
	f1	0.643	0.628	0.947	0.975	f1	0.516	0.471	0.970	0.985		

国立は「日光国立公園」のような固有表現が多いため，

ラベルの平均ベクトルとのコサイン類似度を計算して，

クラス分類を行う

なお，どのような文が正解として予測できなかったかを確認したが，人間でも分類がむずかしそうなものもあった．たとえば，［国立，埼玉，横浜，仙台，札幌］は，おそらくは，サッカーに関連した名前なので，「国立競技場」を意味していると思われるので，「こくりつ」が正解ラベルと思われるが，地名の「くにたち」と予想している．正解ラベルの作成は一苦労であり，何らかの方法を考えなくてはいけない．サンプルの数が増えれば，不均衡データのクラス分類として，再度検証を行ってみたい．

今後も視覚障害当事者の観点から，視覚障害者の情報障害の課題に取り組んでいこうと思っている．

参考文献

[1] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[2] BERT 日本語 Pretrained モデル- KUROHASHI-KAWAHARA LAB http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT%E6%97%A5%E6%9C%AC%E8%AA%9E Pretrained%E3%83%A2%E3%83%87%E3%83%AB

[3] 概要 現代日本語書き言葉均衡コーパス（BCCWJ） http://pj.ninjal.ac.jp/corpus_center/bccwj/

[4] Mikolov, Toma, Wen-tau Yih, and Geoffrey Zweig. "Linguistic regularities in continuous space word representations." Proceedings of the 2013 conference

[5] Welcome to janome's documentation! (Japanese) https://mocobeta.github.io/janome/

[6] GitHub - neologd/mecab-ipadic-neologd: Neologism dictionary based on the language resources on the Web for mecab-ipadic https://github.com/neologd/mecab-ipadic-neologd

[7] scikit-learn: machine learning in Python ? scikit-learn 0.22.1 documentation https://scikit-learn.org/stable/