SVM を用いた BCCWJ における同形異音語の読み推定

小林汰一郎
茨城大学工学部情報工学科
17t4036s@vc.ibaraki.ac.jp

古宮嘉那子
茨城大学理工学研究科工学野情報科学領域
kanako.komiya.nlp@vc.ibaraki.ac.jp

1 はじめに

日本語には同形異音語と呼ばれるものがある。これは文字通り「同じ字形で異なる読みを持つ語彙」のことであり、例として「強請る」などがある。これは「ユスル」と「ネダル」という 2 つの読みがあるが、これらは明らかに異なる意味であり、文章によって適切に読み分ける必要がある。日本語話者であれば、どちらの読みが正しいのか文脈から判断することはさほど難しくない。また、3 種類以上の読みを持つ同形異音語も存在する。「辛い」などがその最たる例である。これは「カライ」「ツライ」「ヅライ」のように読むことができる。「カライ」と「ツライ」が違う意味であることは言うまでもないが、文章によっては「ツライ」という読みに連濁が発生し、「ヅライ」と読むこともある。この連濁を無視して文章を読むことは不自然であることから、「ツライ」と「ヅライ」にも読み分けが必要であることが分かる。

このように、同形異音語の読み分けは文章の意味を理解する上で必須である。ただ、その読み分けには文脈の理解が必要であると予想され、機械が読み分けることは困難である。そこで、文脈を考慮できる分散表現の一種である BERT など複数のベクトル化手法を用いて単語をベクトル化し、それらを入力とした SVM を用いて同形異音語の読み分類を行った。

2 関連研究

本研究は語義曖昧性解消の一種と捉えられる。語義曖昧性解消とは、複数の語義を持つ単語を一意に識別するタスクのことである。語義曖昧性解消は、頻出の単語のみを対象語とする Lexical Sample Task と文中の全単語を対象とする All-words Word Sense Disambiguation（[1]、[2] など）に分けられる。本研究は、Lexical Sample Task の一種として捉えられる。日本語の Lexical Sample Task には、SENSEVAL-2 Japanese dictionary task（[3]）や SemEval-2010 Japanese WSD Task（[4]）などがある。

3 ベクトル化手法

SVM を用いて同形異音語の読み分類をする際に、以下のようなベクトル化の手法を試みた。

1. one-hot ベクトル
2. word2vec
3. BERT

本節では、これらの手法について

- 「私はその辛いカレーをその時食べた。」(以下文章 1)
- 「あなたはその辛い出来事を忘れてはいけない。」(以下文章 2)

を例にとって説明する。

3.1 one-hot ベクトルを用いた手法

one-hot ベクトルとは、対応する素性だけを 1 とし、その他は全て 0 としたベクトルのことである。本研究で利用した作成方法は以下の通りである。

1. 同形異音語の周辺 6 単語を抽出する
2. 抽出した単語に数値を割り当てる
3. one-hot ベクトルに変換する
4. 同形異音語 1 単語毎にベクトルを連結する

まず 1. では文章中の同形異音語の前後 3 単語ずつを抽出する。今回の例で言えば、文章 1 と文章 2 から抽出される単語はそれぞれ「私」「は」「その」「カレー」「を」「その」と「あなた」「は」「その」「出来事」「を」「忘れる」の計 12 単語である。

2. では抽出された単語の辞書を作成し、素性とする。ただし、同じ単語でも、出現した場所によって異なる素性とする。今回で言えば、文章中の同形異音語「辛い」の 1 つ前に出現した「その」と 3 つ後に出現した「その」とは区別して素性とする。

3. では 2. で作成した素性に対して、周辺単語が属する素性を 1 とし、その他は全て 0 とする。

4. では 3. で作成した one-hot ベクトルを単語毎に連結させ、1 つのベクトルとする。

この手法では、文章数によってベクトルの次元が

大きく変化しうる。今回の実験では抽出された文章は 1,441,636 文となり、162,314 次元のベクトルとなった。

3.2 word2vec を用いた手法

本研究では nwjc2vec[1] [5] を利用した。nwjc2vec は国語研日本語ウェブコーパス (NWJC)([6]) から学習された単語の分散表現データであり、word2vec[2] を基に構築された。単語分散表現とは、単語を高次元のベクトルに変換する技術の総称で、word2vec を皮切りに世界中で研究されている。word2vec とは Mikolov ら [7] が 2013 年に提案した手法で、例えば、「王様」を表すベクトルから「男性」を表すベクトルを引いて、更に「女性」を表すベクトルを足すと、「女王」を表すベクトルになるように、単語間で構成性を持つのが大きな特徴である。また、word2vec では同じ表層を持つ単語は同じベクトルに変換される。例えば多義語でも同じベクトルに変換される。これは同形異音語も同様であるため、同形異音語自身の word2vec を素性としても読み分類には役立たない。

今回使用した nwjc2vec の学習の詳細は以下の通りである。

- アルゴリズム : C-BOW
- 次元数 : 200
- ウィンドウ幅 : 8
- ネガティブサンプリングに使用した単語数 : 25
- 反復回数 : 15

one-hot ベクトルの時と同じように、ベクトル化の対象となる単語は文章中の同形異音語の周辺 3 単語ずつである。今回の例で言えば「私」「は」「その」「カレー」「を」「その」と「あなた」「は」「その」「出来事」「を」「忘れる」の計 12 単語である。各単語ベクトルを連結する際、出現した順番に連結することで単語の位置を考慮した。nwjc2vec は 1 単語当たり 200 次元のベクトルであるため、6 単語の抽出を行った本実験では、1 文章につき 1,200 次元のベクトルとなった。

3.3 BERT を用いた手法

BERT[8] とは'Bidirectional Encoder Representations from Transformers' の略称で、Jacob Devlin らが 2018 年に発表した分散表現及びそれを利用したモデルである。BERT の大きな特徴として、文脈に依存した出力を行うことが挙げられる。これは、同じ単語でも文脈によって違うベクトルを出力するということである。本研究では nwjc-BERT[3] と呼ばれる、言語資源協会より公開されている事前学習モデルを利用した。nwjc-BERT は、国立国語研究所コーパス開発センター超大規模コーパスプロジェクトで整備されたウェブテキストコーパスから訓練された、48,914 語彙素からなる BERT モデルであり、同データの 226 億語から、表層形でない UniDic 語彙素に基づいて訓練された。次元数は 1 単語当たり 768 次元である。

nwjc-BERT を用いてベクトル化する際には、同形異音語自身をベクトル化の対象とした。上述の通り、BERT は文脈を考慮し、それ故に同じ単語であっても文章によって異なるベクトルを出力できる。この特性を利用し、周辺単語を抽出してベクトル化せず、文章中の同形異音語自身をベクトル化の対象とした。ベクトル化の際は左右 6 単語を文脈として入力した。

4 実験

実験には現代日本語書き言葉均衡コーパス (以下 BCCWJ)[4] を利用した。

BCCWJ に含まれているテキストとしては以下のような種類があり、今回の実験では全てのジャンルを対象とした。

- 出版サブコーパス
  - 書籍
  - 雑誌
  - 新聞
- 図書館サブコーパス
  - 書籍
- 特定目的サブコーパス
  - 白書
  - 教科書
  - 広報誌
  - ベストセラー
  - Yahoo!知恵袋
  - Yahoo!ブログ
  - 韻文
  - 法律
  - 国会会議録

実際に BCCWJ に含まれていた同形異音語は 71 種存在した (表 1) が、単語数が少なすぎるものについては実験の対象外とした。具体的には、閾値を 50 と設定し、合計単語数が閾値を下回った単語 (表 1 下部の「巨頭」「泡沫」「竹馬」) を実験の対象外とした。

サポートベクトルマシン (SVM) には scikit-learn[5] の

1) https://www.anlp.jp/proceedings/annual_meeting/2017/pdf_dir/E1-5.pdf
2) https://github.com/svn2github/word2vec
3) https://www.gsk.or.jp/catalog/gsk2020-e/
4) https://pj.ninjal.ac.jp/corpus_center/bccwj/
5) https://scikit-learn.org/stable/

表 1 単語リスト

語彙素	読み 1	読み 2	読み 3	読み 4	読み 5	読み 6	単語数
一味	ひとあじ	いちみ					516
一声	ひとこえ	いっせい					267
一目	ひとめ	いちもく					1545
一見	いっけん	いちげん					1983
一途	いっと	いちず					544
上品	じょうひん	じょうぼん					941
上手	うわて	かみて	じょうず	じょうて			3661
下品	げひん	げぼん					289
下手	したて	しもて	へた	げしゅ			2453
人気	ひとけ	にんき					9066
出所	でどころ	しゅっしょ					1233
判官	ほうがん	じょう					205
名代	なしろ	みょうだい	なだい				67
呪い	のろい	まじない					663
外面	そとづら	がいめん					150
寒気	さむけ	かんき					314
居る	いる	おる					1203772
市場	いちば	しじょう					14540
強請る	ゆする	ねだる					443
弾き	はじき	ひき					175
弾く	はじく	ひく					2191
心中	しんじゅう	しんちゅう					669
怒る	おこる	いかる					5493
悪阻	つわり	おそ					394
惚ける	ほうける	ぼける					569
捲る	まくる	めくる					2858
摘む	つむ	つまむ					1192
擦り付ける	こすりつける	すりつける	なすりつける				293
方々	かたがた	ほうぼう					4322
早々	はやばや	そうそう					1092
最中	さなか	もなか	さいちゅう				2078
止め	とどめ	とめ	やめ	どどめ			1115
止める	とめる	やめる					21783
正気	しょうき	せいき					291
正面	まとも	しょうめん					4992
気質	かたぎ	きしつ					544
汚れる	けがれる	よごれる					1839
津々	つつ	しんしん					233
滑り	すべり	ぬめり					344
漢人	あやひと	かんじん					86
潜る	くぐる	もぐる					1650
然も	しかも	さも					15333
物心	ものごころ	ぶっしん					245
生物	なまもの	せいぶつ					5960
白鳥	しらとり	はくちょう					443
目下	めした	もっか					472
空く	あく	すく					3977
素性	すじょう	そせい					290
素振り	すぶり	そぶり					415
細々	こまごま	ほそぼそ					313
経緯	いきさつ	けいい					2228
脅かす	おどかす	おびやかす					925
色紙	いろがみ	しきし					245
艶やか	あでやか	つややか					351
見える	みえる	まみえる					35418
見物	みもの	けんぶつ					1248
解く	とく	ほどく					2307
軽々	かるがる	けいけい					143
辛い	からい	つらい	づらい				8060
逸れる	それる	はぐれる					444
道程	みちほど	どうてい					555
避ける	さける	よける					6586
鈍い	のろい	にぶい					893
長尾	ながお	ちょうび					60
開く	あく	ひらく					19198
難い	かたい	がたい	にくい				12485
頭	あたま	かしら	かぶり	こうべ	どたま	とう	26123
馬頭	ばとう	めず					64
巨頭	きょとう	ごんどう					30
泡沫	うたかた	ほうまつ					43
竹馬	たけうま	ちくば					47

SVC 及び LinearSVC を利用した。ベクトル化手法毎のそれぞれ使用したカーネルは表 2 の通りである。

表 2 使用したカーネル

one-hot	linear
word2vec	linear
BERT	linear poly rbf sigmoid

実験結果のマクロ平均、マイクロ平均を表 3 及び表 4 に示す。また、各単語において、(最頻出読み数)/(全読み数) を BASELINE として設定した。この BASELINE のマクロ平均とマイクロ平均もそれぞれ表 3 と表 4 に示す。

表 3 マクロ平均

	linear	poly	rbf	sigmoid
one-hot	90.69			
word2vec	83.68			
BERT	79.30	82.25	82.04	81.14
BASELINE	79.79			

表 4 マイクロ平均

	linear	poly	rbf	sigmoid
one-hot	96.01			
word2vec	85.40			
BERT	87.45	89.29	89.17	88.88
BASELINE	87.53			

5 考察

表 3、表 4 から、one-hot ベクトルが最も良く、BERT が最も悪いという結果が読み取れる。one-hot ベクトルには、定型的な表現やルールが存在するものの分類には強いという特徴がある。つまり、one-hot ベクトルが最も良かった要因としては、読みの分類には前後の単語によるシンプルなルールがあるからだと考えられる。

また、one-hot ベクトルには未知語が存在しない。これは、素性として登録されていない単語が出てきたら、新たに対応した素性を作成するというアルゴリズム故である。それに対して、nwjc2vec や nwjc-BERT はあらかじめ登録された単語しかベクトル化できない。これによって、one-hot ベクトルが単語分散表現よりも良い結果が得られたのではないかと考えられる。

そこで nwjc2vec と nwjc-BERT に対して未知語を検出した結果、その数は表 5 のようになった。

このように、nwjc-BERT では未知語の割合が 1/6 程度を占めているが、これは 1 つの文章につき 1 単語程度未知語が発生している計算となる。また、nwjc2vec と比べても未知語が 250 倍程度ある。そのため、one-hot ベクトルと nwjc2vec に比べて、nwjc-BERT が最も悪い結果になったと考えられる。

表 5 未知語の数とその割合

ベクトル化手法	単語数	割合
nwjc2vec	5495	0.06
BERT	1375764	15.9

6 おわりに

本稿では BCCWJ から同形異音語を抽出し、その周辺単語や同形異音語自身を素性として、読みの分類を SVM で行った。ここで、周辺単語を素性とする時は前後 3 単語のみを素性とした。

単語のベクトル化手法には one-hot ベクトルと単語分散表現を用い、単語分散表現では nwjc2vec と nwjc-BERT を用いた。これらのベクトル化された単語を SVM の入力とし、one-hot ベクトル及び nwjc2vec では linear カーネルのみを、nwjc-BERT では linear、poly、rbf、sigmoid の 4 種のカーネルを試した。

結果としてはマクロ平均・マイクロ平均共に one-hot ベクトルが最も高く、BERT が最も低いという結果になった。

今後の展望として、ベクトル化手法として BERT を用いる場合、今後は 1 文全てを素性として実験しようと考えている。BERT は文脈を考慮して単語をベクトル化できるため、ベクトル化の際に与える単語をよりに広げることで、今回の実験結果より高い精度が得られると考えられる。より高い精度が得られれば、転移学習による他のタスクの解決にもつながると考えている。

その他にも one-hot ベクトルと BERT の複合ベクトルを入力とするなど、ベクトル化手法の工夫も考えている。また、非線形分類を行うために MLP など、SVM 以外を用いた分類手法を取り入れることも視野に入れている。

謝辞

本研究は、茨城大学の特色研究加速イニシアティブ個人研究支援型「自然言語処理、データマイニングに関する研究」に対する研究支援　および JSPS 科研費 17KK0002、および 18K11421 の助成を受けたものです。

参考文献

[1] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pp. 189–196,

1995.

[2] 浅原正幸 佐々木稔 新納浩幸鈴木類. 概念辞書の類義語と分散表現を利用した教師無し all-words WSD. 自然言語処理 Vol26 No2., 2019.

[3] Kiyoaki Shirai. Senseval-2 japanese dictionary task. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pp. 33–36, 2001.

[4] Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono. On semeval-2010 japanese wsd task. *Information and Media Technologies*, Vol. 6, No. 3, pp. 730–744, 2011.

[5] 新納浩幸, 浅原正幸, 古宮嘉那子, 佐々木稔. nwjc2vec: 国語研日本語ウェブコーパスから構築した単語の分散表現データ. 自然言語処理, Vol. 24, No. 5, pp. 705–720, 2017.

[6] Masayuki Asahara, Kikuo Maekawa, Mizuho Imada, Sachi Kato, and Hikari Konishi. Archiving and analysing techniques of the ultra-large-scale web-based corpus project of ninjal, japan. *Alexandria*, Vol. 25, No. 1-2, pp. 129–148, 2014.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. ICLR, 2013.

[8] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pp. 5753–5763, 2019.