BERTを利用した教師あり学習による語義曖昧性解消

著者	曹 鋭, 田中 裕隆, 白 静, 馬 ブン, 新納 浩幸
雑誌名	言語資源活用ワークショップ発表論文集
巻	4
ページ	273-279
発行年	2019
URL	http://doi.org/10.15084/00002578

BERT を利用した教師あり学習による語義曖昧性解消

曹鋭 (茨城大学大学院理工学研究科情報工学専攻) *
田中裕隆 (茨城大学工学部情報工学科) †
白静 (茨城大学大学院理工学研究科情報工学専攻) ‡
馬ブン (茨城大学大学院理工学研究科情報工学専攻) §
新納浩幸 (茨城大学大学院理工学研究科情報工学専攻) ¶

Word sense disambiguation using supervised learning with BERT

Cao Rui (Graduate School of Science and Engineering, Ibaraki University)
Hirotaka Tanaka (Department of Computer and Information Sciences, Ibaraki University)
Bai Jing (Graduate School of Science and Engineering, Ibaraki University)
Ma Wen (Graduate School of Science and Engineering, Ibaraki University)
Hiroyuki Shinnou (Graduate School of Science and Engineering, Ibaraki University)

要旨

BERT は Transformer で利用される Multi-head attention を 12 層（あるいは 24 層）積み重ねたモデルである．各層の Multi-head attention は，基本的に，入力単語列に対応する単語埋め込み表現列を出力している．つまり BERT は入力文中の単語に対する埋め込み表現を出力しているが，その埋め込み表現がその単語の文脈に依存した形になっていることが大きな特徴である．この点から BERT から得られる多義語の埋め込み表現を，その多義語の語義曖昧性解消ための特徴ベクトルとして扱えると考えられる．実験では京都大学が公開している日本語版 BERT 事前学習モデルを利用して，上記の手法を SemEval-2 の日本語辞書タスクに対する適用し，高い正解率を得た．

1. はじめに

本論文では語義曖昧性解消 (Word Sense Disambiguation, 以下 WSD) における対象単語の特徴ベクトルとして BERT (Devlin et al. (2018)) から得られる単語の埋め込み表現を利用することを提案する．

WSD とは文中の多義語の語義を識別する処理である．例えば単語「犬」は，通常，(a) 動物の犬，(b) スパイ，の 2 つの語義を持ち，「犬」を含む文「あいつは警察の犬だ」が与えられたときに，文中の「犬」の語義が (a) か (b) かを識別する処理が WSD である．WSD は分類問

* 18ND305G@vc.ibaraki.ac.jp
† 16T4032N@vc.ibaraki.ac.jp
‡ 19ND301R@vc.ibaraki.ac.jp
§ 19ND302H@vc.ibaraki.ac.jp
¶ hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

題そのものであり，教師あり学習を用いることで解決できる．この場合，ラベル付き訓練データ（語義タグ付きの対象単語に対する用例）を用意しなくてはならない点が大きな課題としてあり，半教師あり学習や教師無し学習が試みられている．ただしどのような学習手法を用いるにしても，共通して問題となるのは文中の WSD の対象単語をどのような特徴ベクトルで表現するかである．従来は対象単語の周辺単語の字面，品詞，係り受け，シソーラス情報などを素性とし，それらを one-hot-vector によって特徴ベクトルを作成していた．本論文では対象単語の特徴ベクトルとして，BERT からの得られる対象単語の埋め込み表現を利用する．

BERT とは事前学習モデルの一つであり，入力文（単語列）に対して，その単語列に対する埋め込み表現列を出力する．各単語の埋め込み表現はその単語の入力文内の文脈に依存した形となっている．つまり BERT から得られる単語の埋め込み表現はその単語の意味を表現していると考えられる．このためこの埋め込み表現を直接 WSD に利用できるはずである．

実験では京都大学が公開している日本語版 BERT 事前学習モデルを利用して，上記の手法を SemEval-2 の日本語辞書タスク (Okumura et al. (2011)) に対して適用した．結果，非常に高い精度を得ることができた．

2. 関連研究

WSD は分類問題であるため，教師あり学習による解決できる．教師あり学習を行う際に最も重要な問題は，データをどのような特徴ベクトルとして表現するかである．WSD の場合，データは対象単語を含む文であり，従来は対象単語の前後数単語の情報を素性として用いて，それらを one-hot-vector によって特徴ベクトルを作成していた．単語の情報としては，字面，原形，品詞などがあるが，これらだけでは不十分であり，通常，単語のシソーラスの情報を用いている．

通常の素性の他にシソーラスの情報を用いるアプローチは有力だが，これは原理的には対象単語の周辺文脈をある空間に写影し，その空間上の点を素性として利用している形である．この場合，写影される点が離散的であるために，粒度の問題が生じているが，連続的なものにすれば粒度の問題は生じない．これは名詞間距離を設定することでも可能であるが，より精緻に文脈を表現するためにトピックモデルを利用したり (Cai et al. (2007))，分散表現を利用するアプローチが試みられている (Sugawara et al. (2015)Iacobacci et al. (2016))．

単語の分散表現は大規模コーパスと構築ツールである word2vec や GloVe などを利用して作成できる．分散表現が自然言語処理システムで有用なのは，それが従来の単語の特徴ベクトルよりも，より適切にその単語の意味を表現しているからである．しかし分散表現は単語に対して 1 つに固定されているため，多義語に対しても分散表現は 1 つである．つまり分散表現が単語の意味を表していたとしても，WSD に直接は利用できない．この点から語義の分散表現を構築する研究がなされている (Neelakantan et al. (2015)Chen et al. (2014))．語義の分散表現が構築できれば，対象単語の周辺文脈ベクトル(1)との距離から WSD が行える．

近年は上記を更に発展させ，言語の事前学習モデルが WSD に使われている．事前学習モデ

(1) 例えば，周辺単語の分散表現の和で求められる．

ルは大規模なコーパスからあらかじめ学習させた状態の言語のモデルである．様々なモデルの形態があるが，標準的にはそのモデルを利用して，入力単語列をその埋め込み表現列へ変換する．OpenAI GPT (Radford et al. (2018)) はニューラルネット翻訳の Transformer (Vaswani et al. (2017)) の decoder 部分を利用した言語モデルである(2)．ELMo (Peters et al. (2018)) は文脈を考慮した単語の分散表現を導くモデルである．実体は 2 層の双方向 LSTM であり，大規模コーパスを利用して言語モデルを学習する．BERT は OpenAI や ELMo の改良版と位置づけられる事前学習モデルである．

3. BERT を用いた WSD

3.1 BERT

BERT の基本のパーツは Multi-head attention である．Multi-head attention は n 単語埋め込み表現列を入力として，各埋め込み表現をより適切なものに変換して出力する．つまり出力は変換された n 単語埋め込み表現列である．

Multi-head attention の概略を述べる．基本は self attention なので Q, K, V の 3 組が入力である．今，単語埋め込み表現が m 次元であったとする．Multi-head attention では m 次元ベクトルを $d_k(= m/k)$ 次元に圧縮する線形変換器を Q, K, V それぞれに対して用意する．Q, K, V の実体は $d_k \times d_k$ の線形変換行列である．Multi-head attention の入力は n 個の m 次元ベクトルであるが，これが先の圧縮機で $n \times d_k$ の行列 X に変換され，Q, K, V に渡され $n \times d_k$ の行列 XQ, XK, XV ができる．これらを Q', K', V' とおき，以下の式(3)により self attention を行う．

$$\texttt{softmax}\left(\frac{Q'K'^T}{\sqrt{d_k}}\right)V'$$

これは $n \times d_k$ の行列である．上記の処理を k 個並行して行うと，$n \times d_k$ の行列が k 個作成され，これらを横に連結することで，$n \times m$ の行列が作成できる．これを更に同次元に線形変換することで Multi-head attention の出力が作られる．

BERT はこの Multi-head attention を 12 層（あるいは 24 層）重ねたモデルである．結局，BERT は n 単語埋め込み表現列を入力とし，それをより文脈に合った n 単語埋め込み表現列に変換していると捉えることができる．

3.2 feature based の利用

BERT を利用する場合，大きく 2 つの利用法がある．一つは fine tuning である．これは BERT が出力する情報を，タスクを解決するためのネットワークの入力とし，BERT を含めたネットワーク全体を学習の対象とするものである．この場合，事前学習モデルの部分は既に大量のデータから学習できた形となっているため，比較的少量のデータを用いるだけで，連結したネットワークを学習できる．BERT のもう一つの利用法は feature based のものである．これは BERT が出力する情報を，目的のタスクを解くための素性として利用するものである．

(2) 言語モデルは一種の事前学習モデルである．
(3) Scaled Dot-Product Attention

本論文の手法は BERT の出力する WSD の対象単語の埋め込み表現を WSD の特徴ベクトルとしているので feature based である．

4. 実験

4.1 実験データ

WSD のデータとしては SemEval-2 の日本語辞書タスクのデータを用いる．これは WSD の対象単語が 50 個あり，対象単語毎に訓練用例とテスト用例が 50 用例づつ存在する．

用いた日本語 BERT 事前学習モデルは京都大学黒橋・河原研究室が以下のサイトで公開しているものを使用する．

http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT%E6%97%A5%E6%9C%AC%E8%AA%9EPretrained%E3%83%A2%E3%83%87%E3%83%AB

各用例を Juman++ で単語分割し，その単語列を上記 BERT にかけ，出力された単語埋め込み表現列から，WSD の対象単語の埋め込み表現を取り出し，これを WSD の特徴ベクトルとした．

4.2 分類器の学習

学習には 2 層のニューラルネットワークを用いた．入力層は特徴ベクトルの次元数 768 のユニットを持ち，出力層は対象単語の語義数のユニットを持つ (図 1 参照)．訓練用例 50 個で学習し，100 epoch で学習を止めて，そのときに得られるニューラルネットワークを分類器とした．

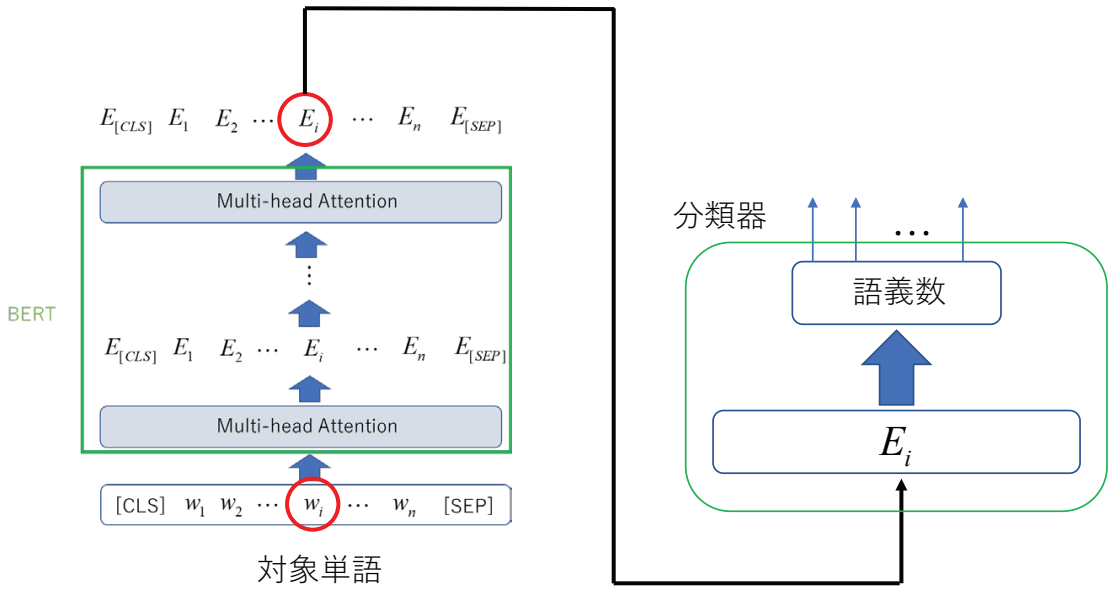

図 1 BERT による WSD

4.3 実験結果

各対象単語に対して学習できた分類器を用いて，50 個のテスト用例の語義を識別し，正解率を測る．実験結果を表 1 に示す．表中の BERT(2) が本手法を表す(4)．また表中の Standard は WSD で従来の用いられてきた標準的な特徴ベクトルと SVM を利用した識別結果である．本手法の正解率が高いことが確認できる．

表 1　実験結果 (正解率)

対象単語	Standard	BERT(2)	BERT(3a)	BERT(3b)
相手	0.82	0.76	0.82	0.82
会う	0.88	0.86	0.66	0.66
上げる	0.42	0.64	0.38	0.38
与える	0.72	0.86	0.58	0.58
生きる	0.94	0.94	0.94	0.94
意味	0.70	0.66	0.54	0.54
入れる	0.74	0.80	0.72	0.72
大きい	0.94	0.94	0.94	0.94
教える	0.18	0.74	0.18	0.18
可能	0.72	0.72	0.56	0.56
考える	0.98	0.98	0.98	0.98
関係	0.98	0.96	0.78	0.78
技術	0.84	0.84	0.84	0.84
経済	0.98	0.98	0.98	0.98
現場	0.80	0.84	0.78	0.78
子供	0.56	0.66	0.36	0.64
時間	0.90	0.82	0.90	0.90
市場	0.72	0.70	0.70	0.70
社会	0.86	0.86	0.86	0.86
情報	0.84	0.88	0.84	0.84
進める	0.88	0.96	0.32	0.32
する	0.46	0.82	0.42	0.42
高い	0.86	0.88	0.86	0.86
出す	0.46	0.60	0.28	0.56
立つ	0.54	0.62	0.52	0.52
強い	0.92	0.94	0.92	0.92
手	0.78	0.78	0.78	0.78
出る	0.72	0.72	0.60	0.60
電話	0.94	0.94	0.56	0.56
取る	0.46	0.44	0.26	0.28
乗る	0.80	0.88	0.50	0.50
場合	0.90	0.86	0.86	0.86
入る	0.60	0.60	0.50	0.50
はじめ	0.98	0.98	0.96	0.98
始める	0.78	0.86	0.78	0.78
場所	0.96	0.96	0.96	0.96
早い	0.82	0.80	0.64	0.64
一	0.92	0.92	0.92	0.92
開く	0.90	0.92	0.90	0.90
文化	0.98	0.98	0.98	0.98
他	1.00	1.00	1.00	1.00
前	0.84	0.84	0.62	0.62
見える	0.54	0.70	0.52	0.52
認める	0.82	0.86	0.70	0.70
見る	0.80	0.80	0.80	0.80
持つ	0.68	0.82	0.68	0.68
求める	0.76	0.76	0.76	0.76
もの	0.88	0.88	0.88	0.88
やる	0.94	0.94	0.94	0.94
良い	0.24	0.70	0.24	0.24
平均	0.77	0.82	0.70	0.71

(4) BERT(3a) と BERT(3b) は考察の章を参照.

5. 考察

本実験では学習に利用したモデルは 2 層の単純なモデルある．これを 3 層にした場合の実験を行った．中間層は 50 に固定した．実験の結果を表 1 に示す．表中の BERT(3a) と BERT(3b) がその結果である．BERT(3a) は BERT(2) と同じく 100 epoch の学習後に得られたモデルの結果であり，BERT(3b) は 100 epoch 内でも最も正解率が高いモデルを洗濯した場合の結果である．

2 層の BERT(2) と比較すると，正解率はかなり悪くなっている．Standard よりも劣る．これは BERT の出力である単語の埋め込み表現は，十分に意味を表現しており，WSD の利用に関しては下手に加工するよりも，そのまま利用する方がよいことを示していると考えられる．

6. おわりに

本論文では WSD における対象単語の特徴ベクトルとして BERT から得られる単語の埋め込み表現を利用することを提案した．

京都大学が公開している日本語版 BERT 事前学習モデルと SemEval-2 の日本語辞書タスクデータを用いた実験では，非常に高い正解率を示した．本実験において分類器の学習を簡易にすませていることを考慮すると，BERT から WSD の特徴ベクトルを得る有用性が示されたと言える．今後は fine tuning の利用，WSI や all-word WSD への適用を試みたい．

文　献

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*.

Manabu Okumura, Kiyoaki Shirai, Kanako Komiya, and Hikaru Yokono (2011). “On SemEval-2010 Japanese WSD Task.” 自然言語処理, 18:3, pp. 293–307.

Junfu Cai, Wee Sun Lee, and Yee Whye Teh (2007). “Improving word sense disambiguation using topic features.” *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pp. 1015–1023.

Hiromu Sugawara, Hiroya Takamura, Ryohei Sasano, and Manabu Okumura (2015). “Context representation with word embeddings for wsd.” *Conference of the Pacific Association for Computational Linguistics*, pp. 108–119., Springer.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli (2016). “Embeddings for word sense disambiguation: An evaluation study.” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 897–907.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum (2015). “Efficient non-parametric estimation of multiple embeddings per word in vector space.”

arXiv preprint arXiv:1504.06654.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun (2014). "A unified model for word sense representation and disambiguation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1025–1035.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever (2018). "Improving language understanding by generative pre-training." *Technical report, OpenAI.*

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." *Advances in neural information processing systems*, pp. 5998–6008.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep Contextualized Word Representations." *NAACL-2018*, pp. 2227–2237.