

語義の例文を使用した語義曖昧性解消の有効性分析

関谷 洸 佐々木 稔
茨城大学工学部情報工学科
{18t4052h, minoru.sasaki.01}@vc.ibaraki.ac.jp

概要

本研究は、語義曖昧性解消システムである BEM [1]において、WordNet3.0 [2]に表記されている語義の例文を使用することの有効性について分析する。具体的には、BEM に語義の例文を入力し、出力される単語埋め込み表現列の中から、[CLS]ベクトルまたは対象単語ベクトルを取り出し、語義の埋め込み表現として使用する。実験では、開発セットを含む六つの評価セットのうち、五つの評価セットで F 値が低下し、一つの評価セットで F 値が向上した。

1 はじめに

語義曖昧性解消とは、文中にある多義語が、どの語義を表しているのかを識別するタスクである。多くの語義曖昧性解消システムは、学習データの不均衡により、出現頻度の高い語義 (MFS) に対しては精度が高いが、出現頻度の低い語義 (LFS) に対しては精度が低いという問題がある。これまでの研究で、語義曖昧性解消システムに辞書の語義定義文などの語彙情報を組み込むことで精度が向上することが示されている [3][4]。語義定義文は LFS にも効果的であることが分かっているが、依然として MFS と LFS の間には大きな差がある。Blevins らが提案した BEM では、複数の事前学習されたエンコーダを共同でファインチューニングすることで、MFS と LFS の間にある大きな精度差を大幅に縮小した。しかし、BEM では、システムに組み込む語彙情報が語義定義文しか使用されていないため、辞書の情報をさらに加えることで語義の特徴を効果的に表現できる可能性がある。

そこで本研究では、BEM に新しい語彙情報として語義の例文を組み込む手法を提案し、語義曖昧性解消における語義の例文の有効性を検証する。

2 関連研究

Blevins らによる BEM は、対象単語と周囲の文脈

を表現する context encoder と、語義定義文を表現する gloss encoder で構成され、対象単語とその語義を同じ埋め込み空間で表現している。この二種類のエンコーダは、事前学習済みモデルである BERT [5]で初期化され、共同でファインチューニングされる。この手法は、先行研究である Raganato ら [6]が示した英語の All-words Word Sense Disambiguation の結果を上回る。本研究では、BEM において、新しい語義の埋め込み表現として語義の例文の埋め込みを作成し、モデルの学習を行った。

BERT とは、Google の Jacob Devlin らが提案した自然言語処理モデルである。BERT は、文章を双方向から学習することが可能であるため、従来のモデルとは違い、文脈に依存した出力を行うことができる。

3 BEM

BEM のモデル構造を図 1 に示す。

BEM は、対象単語と周囲の文脈を表現する context encoder と、語義定義文を埋め込む gloss encoder の二つの独立したエンコーダによって構成されている。これらのエンコーダは、各トークンが正しい語義の表現の近くに埋め込まれるように学習される。また、各エンコーダは、事前学習から取得した語義情報を活用するために、BERT で初期化された deep transformer network である [7][8]。

BEM は、文脈を含む対象単語と語義定義文を独立してエンコードする [9][10]ように設計されており、これらの各モデルは、BERT-base で初期化される。

context encoder T_c は、語義曖昧性解消の対象単語 w を含む用例文 c を入力として受け取る。ここでの c は、 $c = c_0, c_1, \dots, w_i, \dots, c_n$ である。context encoder からは文脈を考慮した単語埋め込み表現列 r が出力される。単語埋め込み表現列 r にある対象単語ベクトル r_{w_i} を対象単語の表現とする。次式で表す r_{w_i} は、 T_c によって出力された i 番目の表現を意味する。

$$r_{w_i} = T_c(c)[i]$$

BERT トークナイザによって複数のサブワードにトークン化された単語については、サブワード部分の平均的な表現によって、単語を表現する。例えば、 j 番目から k 番目までのトークンが i 番目の単語のサブワードに対応する場合、次式のように単語を表現する。

$$r_{wi} = \frac{1}{k-j} \sum_{l=j}^k (T_c(c)[l])$$

gloss encoder T_g は、語義 s を定義する語義定義文 $g_s = g_0, g_1, \dots, g_m$ を入力として受け取る。gloss encoder が出力する単語埋め込み表現列の中から、先頭の表現である [CLS] ベクトルを取り出し、語義の表現 r_s とする。語義の表現 r_s は、次式のように表される。

$$r_s = T_g(g_s)[0]$$

次式で示すように、context encoder によって出力された用例文の対象単語ベクトル r_w と gloss encoder によって出力された対象単語の語義ベクトル r_s のドット積をとることで、対象単語 w の各候補語義 $s \in S_w$ をスコア付けする。 S_w は、対象単語 w のすべての候補語義を意味する。

$$\phi(w, s_i) = r_w \cdot r_{s_i}$$

ここでの i は、 $i = 0, \dots, |S_w|$ である。評価の際、対象単語の語義は各候補語義の中でスコアが最も高い語義に推定される。

対象単語 w の各候補語義のスコアで、損失関数である cross-entropy loss を使用し、BEM を学習する。対象単語と語義のペア (w, s_i) が与えられた場合、システムの損失関数は次式で表わされる。

$$\mathcal{L}(w, s_i) = -\phi(w, s_i) + \log \sum_{j=0}^{|S_w|} \exp(\phi(w, s_j))$$

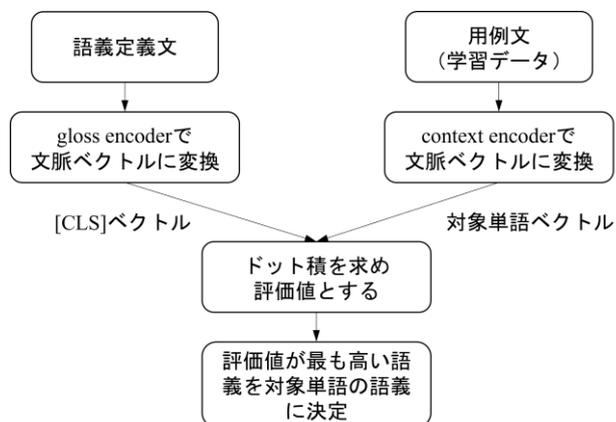


図 1 BEM のモデル構造

4 提案手法

本研究では、BEM に新しく語義の例文を入力し、対象単語の語義推定を行う。主に三種類の語義の例文の表現方法を使用して BEM の学習を行い、作成した三種類のモデルと元の BEM で精度の比較を行う。

4.1 語義の例文の [CLS] ベクトルを使用した語義曖昧性解消

語義の例文の [CLS] ベクトルを使用したモデルの構造を図 2 に示す。

ある語義に対し、複数の例文が存在する場合はその中の一つを取得する。取得した語義の例文を gloss encoder に入力し、出力された単語埋め込み表現列の中から [CLS] ベクトルを取り出す。取り出した [CLS] ベクトルを新しい語義の埋め込み表現として使用する。context encoder によって出力された用例文の対象単語ベクトルと gloss encoder によって出力された語義定義文、語義の例文の二種類の語義ベクトルのドット積をとり、各候補語義のスコア付けをする。各候補語義の中でスコアが最も高い語義を対象単語の語義に推定する。

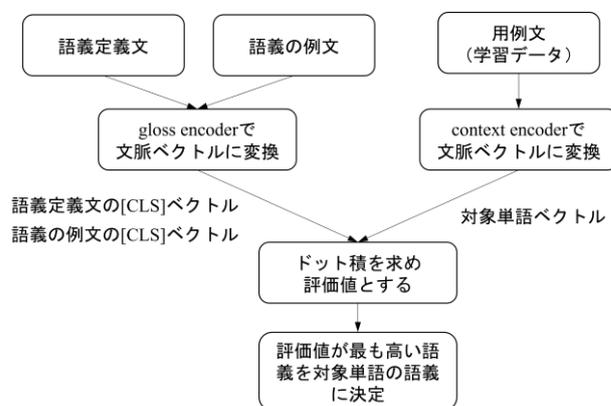


図 2 語義の例文の [CLS] ベクトルを使用したモデル

4.2 語義の例文の対象単語ベクトルを使用した語義曖昧性解消

語義の例文の対象単語ベクトルを使用したモデルの構造を図 3 に示す。

ある語義に対し、文中に対象単語を含む例文が存在する場合はその中の一つを取得する。例えば、対

象単語が「review」の場合、例文中に「review」と同型の単語がある例文を取得する。取得した語義の例文を context encoder に入力し、出力された単語埋め込み表現列の中から対象単語ベクトルを取り出す。取り出した対象単語ベクトルを新しい語義の埋め込み表現として使用する。context encoder によって出力された用例文の対象単語ベクトルと gloss encoder によって出力された語義定義文、context encoder によって出力された語義の例文の二種類の語義ベクトルのドット積をとり、各候補語義のスコア付けをする。各候補語義の中でスコアが最も高い語義を対象単語の語義に推定する。

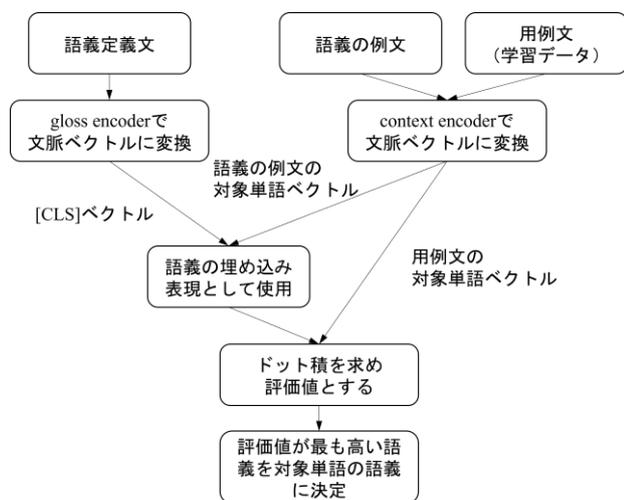


図 3 語義の例文の対象単語ベクトルを使用したモデル

4.3 語義の例文で語義定義文のベクトルを更新した語義曖昧性解消

語義の例文で語義定義文のベクトルを更新したモデルの構造を図 4 に示す。

ある語義に対し、複数の例文が存在する場合はその中の一つを取得する。また、語義に対し例文が存在しない場合、代用として語義定義文を語義の例文として使用する。取得した語義の例文を gloss encoder に入力し、出力された単語埋め込み表現列の中から [CLS] ベクトルを取り出す。取り出した [CLS] ベクトルを新しい語義の埋め込み表現として使用する。語義定義文で各エンコーダをファインチューニングした後、語義の例文でファインチューニングを行い、語義ベクトルを更新する。

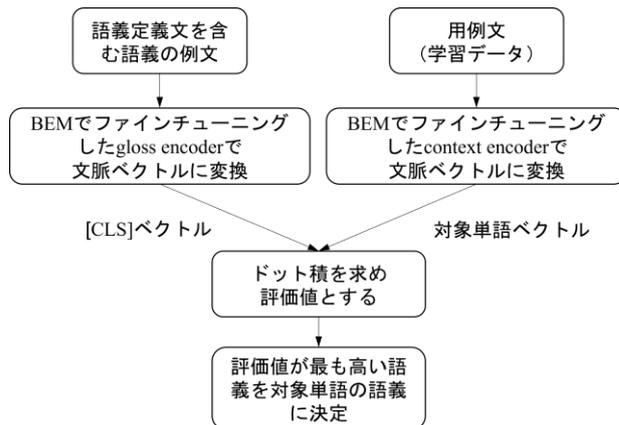


図 4 語義ベクトルを更新したモデル

5 実験

5.1 実験データ

本研究で使用したデータは、Raganato らが作成した WSD フレームワークである [6]。学習データとして、WordNet から手動で語義をアノテーションされた大規模なデータセットである SemCor [11] コーパスを使用する。SemCor は、33,362 個の語義をカバーする 226,036 個のアノテーションされた例を含む。開発セットとして SemEval-2007(SE07) [12] データセットを使用する。評価セットとして、Senseval-2(SE2) [13]、Senseval-3(SE3) [14]、SemEval-2013(SE13) [15]、SemEval-2015(SE15) [16] を使用する。本システムで使用する語義定義文、語義の例文はすべて WordNet3.0 から取得している。

5.2 実験設定

BEM、4.1 節、4.2 節、4.3 節で示したモデルを表 1 にまとめる。これらのモデルの F 値を比較し、語義の例文の有効性を分析する。

表 1 BEM と本研究で提案するモデル

モデル名	説明
BEM	Blevins らが提案したモデル
BEM1	語義の例文の [CLS] ベクトルを使用 (4.2 節)
BEM2	語義の例文の対象単語ベクトルを使用 (4.3 節)
BEM3	語義の例文で語義定義文のベクトルを更新 (4.4 節)

本研究で提案するモデルは、実行時間に制限がある Google Colaboratory で実行している。また、最も F 値が高いモデルのみを保存し学習を再開するため、

モデルによって学習回数が異なる。BEM1 は 14、BEM2 は 6、BEM3 は 17 回学習を行う。

5.3 評価方法

5.1 節で示した実験データを使用し、各モデルの F 値を求める。5.1 節で示した評価セットに加え、開発セットを含むすべての評価セットを連結させた ALL データセットを評価セットとして使用する。

5.4 実験結果

開発セットを含む六つの評価セットでの BEM、BEM1、BEM2、BEM3 の F 値を表 2 に示す。BEM と語義の例文を使用した BEM1、BEM2、BEM3 を比べると、Senseval-3 評価セットを除くすべての開発セット・評価セットで BEM の方が、F 値が高い。Senseval-3 評価セットでは、BEM2 の 78.1% が最高値であり、BEM に比べ 0.7% 向上している。

表 2 各モデルの F 値

	SE07	SE2	SE3	SE13	SE15	ALL
BEM	74.5	79.4	77.4	79.7	81.7	79.0
BEM1	73.8	78.6	76.7	77.4	80.6	77.8
BEM2	73.6	79.3	78.1	78.2	80.3	78.5
BEM3	73.0	77.6	76.2	76.5	80.1	77.0

6 考察

6.1 語義の例文の[CLS]ベクトル

BEM において、語義の例文の[CLS]ベクトルを使用した語義曖昧性解消は、元の BEM による語義曖昧性解消に比べて、すべての開発セット・評価セットで F 値が低下した。このことから、語義の例文の[CLS]ベクトルを語義の埋め込み表現として使用するとシステムに悪影響を与えられられる。

6.2 語義の例文の対象単語ベクトル

BEM において、語義の例文の対象単語ベクトルを使用した語義曖昧性解消は、開発セットと一つの評価セットを除き、語義の例文を使用したモデルの中で最も F 値が高い。このことから、BEM で語義の例文を使用する場合、語義の例文を対象単語ベクトルで表現することが最も有効であると言える。対象単語ベクトルが最も良かった要因としては、語義の例文は語義定義文とは違い、語義の意味ではなく、語義の使用方法を表しているためだと考えられる。こ

れにより、文中に語義との類似度が低い単語などがあり、[CLS]ベクトルでは F 値が低くなったと考えられる。また、元の BEM による語義曖昧性解消に比べて、Senseval-3 評価セットでは F 値が僅かに高くなり、それ以外の開発セット・評価セットでは低くなっている。このことから、語義の例文の対象単語ベクトルはシステムに悪影響を与えるが、使用方法を工夫すれば有効であると予想している。例えば、対象単語が「review」の場合、review の過去形である「reviewed」や複数形である「reviews」なども対象単語として使用するという方法が考えられる。

6.3 語義ベクトルの更新

BEM で使用していた語義定義文による語義ベクトルを語義の例文で更新した語義曖昧性解消は、元の BEM による語義曖昧性解消に比べて、すべての開発セット・評価セットで F 値が低下した。このことから、語義定義文による語義ベクトルを語義の例文で更新すると、システムに悪影響を与えられられる。本研究では、語義の例文の[CLS]ベクトルを使用して語義ベクトルを更新しているが、対象単語ベクトルを使用して語義ベクトルを更新するという手法も考えられる。

7 おわりに

本研究では、語義曖昧性解消システムである BEM において、WordNet3.0 から取得した語義の例文を新しい語義の埋め込み表現として使用した。また、BEM と語義の例文を使用した BEM を比較し、語義の例文の有効性の分析を行った。語義の例文を使用すると全体的に性能が低下したが、語義の例文の対象単語ベクトルを使用したモデルでは、Senseval-3 評価セットにおいて、F 値が僅かに上昇することが確認できた。よって、語義の例文は BEM に悪影響を与えるが、語義の例文の使用方法を工夫すれば有効であると予想している。

今後の課題としては、語義の例文の対象単語ベクトルを使用する際、対象単語と同型のものだけでなく、複数形や過去形に変形しているものも対象にするなど、抽出方法の検討が挙げられる。その他にも、多義語の語義の数が多い場合のみ語義の例文を使用するなど、語義の例文の使用法の工夫も考えている。

参考文献

1. BlevinsTerra, ZettlemoyerLuke. Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders. : the 58th Association for Computational Linguistics (ACL2020), 2020. ページ: pp. 1006-1017. .
2. George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
3. Fuli Luo, Tianyu Liu, Zexue He, Qiaolin Xia, Zhifang Sui, and Baobao Chang. 2018a. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1402–1411, Brussels, Belgium. Association for Computational Linguistics.
4. Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018b. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482, Melbourne, Australia. Association for Computational Linguistics.
5. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.
6. Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017b. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110.
7. Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viegas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert. *arXiv preprint arXiv:1906.02715*.
8. Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5300–5309.
9. Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Sackinger, and Roopak Shah. 1994. Signature verification using a” siamese” time delay neural network. In *Advances in neural information processing systems*, pages 737–744.
10. Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.
11. George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics.
12. Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task-17: English lexical sample, srl and all words. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 87–92.
13. Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24.
14. Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.
15. Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231.
16. Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 288–297.