埋め込み表現の意味適応による知識ベース語義曖昧性解消

水木栄
東京工業大学情報理工学院
sakae.mizuki@nlp.c.titech.ac.jp

岡崎直観
東京工業大学情報理工学院
okazaki@c.titech.ac.jp

概要

知識ベース語義曖昧性解消（WSD）の有望な方法論は，文脈依存埋め込みによる埋め込み空間上で対象単語に最も近い語釈文の語義を選ぶことである．本研究では，語彙知識を用いて埋め込み表現を WSD に適応させる手法を提案する．提案手法の鍵は，関連する語義対および語義・用例対を近づけて，無関連な語義対および異義対を遠ざけることである．これらを実現するため，吸引・反発学習および自己学習を用いる．提案手法は知識ベース WSD の最高精度を達成した．また分析により，両学習を併用する有効性を確認した．

1 はじめに

語義曖昧性解消（WSD）とは，文脈を考慮して適切な語義を選択するタスクである．WSD の用途は評判分析 [1, 2]，情報検索 [3]，機械翻訳 [4] がある．本研究は，WordNet [5] の語彙資源のみを用いる知識ベース WSD の精度向上に取り組む．知識ベース WSD は教師あり WSD に対して精度面で劣るが，高コストな語義注釈付き用例文を用いずに済む．

知識ベース WSD の有望な方法論は，文脈依存埋め込みによる最近傍法である [6]．具体的には，BERT [7] などの事前学習ずみ言語モデルを用いて，語釈文の埋め込み—**語義埋め込み**—および，用例文内の対象単語の埋め込み—**文脈埋め込み**—を計算し，文脈に最も近い語義を選択する．最近傍法の鍵は，語釈文と用例文の対応付けである．すなわち埋め込み空間上で正しい語義・用例対を近づけ，WSD に適応させたい．語義注釈付き用例文を用いず，語彙資源のみで対応付けを改善できるだろうか？

既存研究ではふたつの改善策がある．ひとつは語彙知識による埋め込み表現の適応である．SREF [6] は，意味的に関連する語義対の埋め込みを近づけて精度を改善した．しかし関連しない語義対や，語義・用例対は活用していない．もうひとつは文脈情報の拡張である．COE [8] は，評価文書内の隣接文などを用いて最高精度を達成した．しかし，SNS 短文などの文単体には適用できず，汎用性に欠ける．

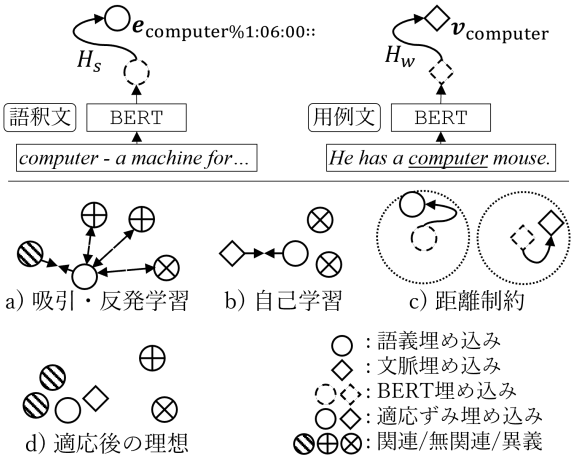

図 1 提案手法の概要．BERT の語義・文脈埋め込みを変換して適応させる（上）．変換関数の最適化は，距離制約のもとで，吸引・反発学習と自己学習を併用する（下）．

本研究は，BERT 埋め込みを変換して WSD に適応させる手法（図 1）に取り組む．本研究の提案の核心は，関連する語義対および語義・用例対を近づけ，関連しない語義対および異義対は遠ざけることである．具体的には，距離制約のもとで，**吸引・反発学習**（図 1-a）および**自己学習**（図 1-b）を行う．前者は語義間の識別性を，後者は語義と用例の擬似的な対応付けをそれぞれ学習する．提案手法の主な新規性は，知識ベース WSD への吸引・反発学習の適用および自己学習の併用である．埋め込み表現の意味適応手法の総称 [9] に準じて，提案手法を Semantic Specialization for WSD (SS-WSD) と呼ぶ．

実験の結果，提案手法の精度は従来の埋め込み表現適応法を上回った．また既存研究で有効とされる最近傍語義探索の経験則を併用すると，文書の情報を用いずに現時点の世界最高精度を達成した．本研究の貢献は次の 2 点である．文単体に適用可能な，吸引・反発学習と自己学習の併用による新手法の提案．知識ベース WSD の最高精度の達成．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

2 既存研究

2.1 知識ベース WSD

知識ベース WSD では古くから語釈文と用例文の類似性が使用されている [10]．また BERT 埋め込みは文脈を考慮して疎粒度の語義を捉えるとの報告がある [11, 12]．かかる背景のもと，SREF [6] は BERT 埋め込み表現による最近傍法および，上位下位語義等との加重平均による語義埋め込み適応の有効性を報告した．これを発展させた COE [8] は，評価文書内の隣接文や共起語義[1)]による文脈埋め込みを併用して最高精度を達成した．本研究は文書情報を用いずに，これらを上回る埋め込み表現適応を提案する．

2.2 埋め込み表現の意味適応

語彙資源に含まれる知識を事前学習ずみの埋め込み表現に注入して意味適応させる手法は Semantic Specialization (SS) と総称される．SS の目的は，文脈類似性に基づく意味的類似度と，語彙資源に基づく詳細な意味関係の統合である [9]．先行研究 [9, 14] は，上位下位・対義などの関係知識を用いて静的な単語埋め込みを意味適応させ，単語間意味関係識別タスクでの有効性を報告した．本研究は，文脈依存埋め込みの意味適応による WSD に取り組む．

3 提案手法

3.1 埋め込み表現の意味適応による WSD

提案手法は，BERT による埋め込みを変換する．

$$\mathbf{v}_w = H_w(\hat{\mathbf{v}}_w) \tag{1}$$

$$\mathbf{e}_s = H_s(\hat{\mathbf{e}}_s) \tag{2}$$

入力 $\hat{\mathbf{v}}_w$ および $\hat{\mathbf{e}}_s$ は BERT を用いて計算した文脈 w および語義 s の埋め込み，出力 $\mathbf{v}_w$ および $\mathbf{e}_s$ は適応ずみ埋め込み，H_w および H_s は変換関数である．訓練時は，変換関数を学習する．具体的には $\mathbf{v}_w$ および $\mathbf{e}_s$ を使って吸引・反発学習および自己学習の損失加重和を最小化する．BERT 自体はファインチューニングしない．埋め込みそのものを適応させるのではなく，変換関数を学習することで，任意の文脈埋め込みを処理できるようにする．推論時は，変換した（適応ずみの）埋め込みを用いて最近傍の語義を選ぶ．具体的には，曖昧性を解消する単語 w および候補語義 $s' \in \mathcal{S}_w$ の埋め込みをそれぞれ変換してから，cosine 類似度が最大の語義 s^* を選択する．

$$s^* = \arg\max_{s' \in \mathcal{S}_w} \rho_{w,s'} \tag{3}$$

$$\rho_{w,s} = \cos(\mathbf{v}_w, \mathbf{e}_s) = \frac{\mathbf{v}_w \cdot \mathbf{e}_s}{\|\mathbf{v}_w\|\|\mathbf{e}_s\|} \tag{4}$$

訓練および推論に必要な語彙資源は WordNet から取得する．語義の識別子は sense key である．BERT による文脈・語義埋め込みの計算方法は，先行研究 [6] にならう（付録 B）．語義埋め込みの計算にはレンマ・同義語・語釈文・例文の連結を使用する．

3.2 変換関数

埋め込みを適応させる変換関数（式 2）は，順伝播型 NN（FFNN）による残差接続で定式化する[2)]．

$$\mathbf{v}_w = H_w(\hat{\mathbf{v}}_w) = \hat{\mathbf{v}}_w + \epsilon\|\hat{\mathbf{v}}_w\|F_w(\hat{\mathbf{v}}_w) \tag{5}$$

$$\mathbf{e}_s = H_s(\hat{\mathbf{e}}_s) = \hat{\mathbf{e}}_s + \epsilon\|\hat{\mathbf{e}}_s\|F_s(\hat{\mathbf{e}}_s) \tag{6}$$

$$F_w(\cdot) = 2\sigma(\mathrm{FFNN}_w(\cdot)) - 1 \tag{7}$$

$$F_s(\cdot) = 2\sigma(\mathrm{FFNN}_s(\cdot)) - 1 \tag{8}$$

σ はシグモイド関数である．ϵ は変換による移動を調整するハイパーパラメータである．具体的には，変換前の BERT 埋め込みからの相対 L2 距離を $\|\mathbf{v}_w - \hat{\mathbf{v}}_w\|/\|\hat{\mathbf{v}}_w\| \leq \epsilon\sqrt{N_d}$ に抑える[3)]．この距離制約の意図は，自己学習の促進である．BERT による類似度特性を適度に保つことで，語義・用例の不正確な対応付けによる精度低下の悪循環を回避する．距離制約の有効性は実験的に検証する（§ 5.2）．

3.3 訓練の目的関数

訓練時の目的関数は，吸引・反発学習損失 L^{AR} および自己学習損失 L^{ST} の加重和である．

$$L = L^{\mathrm{AR}} + \alpha L^{\mathrm{ST}} \tag{9}$$

α は自己学習の重要度を制御するハイパーパラメータである．両学習を併用する動機は，相互補完的な役割にある．吸引・反発学習は語義間の識別性に寄与するが，語義・用例対への教師信号はない．一方で自己学習（§ 3.3.2）は語義・用例対への教師信号を提供するが，単体では BERT が既に対応付けた語義・用例対を強化するのみで新規の情報は限られるはずである．その効果は実験的に検証する（§ 5.1）．

1) 文書内の語義割り当ては一貫する仮説 [13] にもとづく．

2) 層数は 2，活性化関数は ReLU とする．

3) 残差接続 F の出力は各次元要素が [-1,1] に制約されるため．N_d は次元数．bert-large-cased では $N_d = 1,024$.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

3.3.1 吸引・反発学習

吸引・反発学習に用いる語義対は，WordNet の意味関係知識から作成する．具体的には語義 s に対して，関連 $\mathcal{S}_s^P$，異義 $\mathcal{S}_s^N$，無関連 $\mathcal{S}_s^U$ の 3 種類の語義集合を取得する．$\mathcal{S}_s^P$ は同義や上位下位など，意味関係でつながる語義である．正確な定義は先行研究 [6] に従う（付録 A）．$\mathcal{S}_s^N$ は共通のレンマを持つ異なる語義である[4]．$\mathcal{S}_s^U$ は乱択した語義である．語義の統計量を付録表 5 に示す．

吸引・反発損失 L^{AR} の定式化は，対照損失を用いる．具体的には語義 s に対して，関連 $\mathcal{S}_s^P$ を近づけて，異義 $\mathcal{S}_s^N$ および無関連 $\mathcal{S}_s^U$ を遠ざける．まず，乱択ミニバッチ内の語義 $s \in \mathcal{S}^B$ を所与とする．次に s を除いたものを無関連 $\mathcal{S}_s^U = \mathcal{S}^B \setminus \{s\}$ とする．同様に s_p は $\mathcal{S}_s^P$ から 1 個を乱択，$\tilde{\mathcal{S}}_s^N$ は $\mathcal{S}_s^N$ から最大 5 個を乱択する．L^{AR} の定義は以下のとおり[5]．

$$L^{\mathrm{AR}} = -\sum_{s \in \mathcal{S}^B} \ln \frac{\exp(\beta \rho_{s,s_p})}{\sum_{s' \in (\{s_p\} \cup \mathcal{S}_s^U \cup \tilde{\mathcal{S}}_s^N)} \exp(\beta \rho_{s,s'})} \tag{10}$$

$$\rho_{s,s'} = \cos(\mathbf{e}_s, \mathbf{e}_{s'}) \tag{11}$$

3.3.2 自己学習

自己学習に用いる語義・用例対は，WordNet の語義目録から作成する．具体的には，対象単語 w のレンマ・品詞ペアに紐づく語義の集合 $\mathcal{S}_w$ を取得する．

自己学習損失 L^{ST} の定式化は，用例に最も近い語義との類似度を最大化する．すなわち，最近傍語義を擬似的な正解とする学習である．具体的には，用例文に含まれる対象単語 $w \in \mathcal{W}^B$ の候補語義集合 $\mathcal{S}_w$ を取得する．そして単語と語義の埋め込みをそれぞれ変換して cosine 類似度を計算し，候補内での最大値を取る．

$$L^{\mathrm{ST}} = \sum_{w \in \mathcal{W}^B} (1 - \max_{s \in \mathcal{S}_w} \rho_{w,s}) \tag{12}$$

$$\rho_{w,s} = \cos(\mathbf{v}_w, \mathbf{e}_s) \tag{13}$$

最近傍語義は不変ではなく，変換関数の学習につれて変わることに注意されたい．変化させる意図はブートストラップである．すなわち訓練が進捗して精度が向上する過程で，擬似正解の正確性も向上する好循環を期待する．変換前後の距離制約（§ 3.2）はその逆，すなわち悪循環を回避する意図である．

4) すなわち，多義語の語義集合から自身を除いたもの．
5) 距離学習の先行研究 [15, 16] に倣い，$\beta = 64$ に設定した．

3.4 Try-again Mechanism (TaM) 経験則

推論で最近傍語義を選ぶとき，Try-again Mechanism (TaM) なる経験則が有効であると知られている [17, 6, 8]．そこで，提案手法に TaM を併用する効果を報告する．TaM の概要は，候補語義を 2 つに絞り込み，各語義が属する意味カテゴリとの類似度を考慮して再順位付けする処理である．具体的なアルゴリズム（付録 C）は，先行研究 [17] を参照されたい．

4 実験

吸引・反発学習の訓練データとなる語義対は WordNet を用いた（語義数は 206,949）．自己学習の用例文は，正解語義を削除した[6] SemCor コーパス [20] を用いた（単語数は 226,036）．ハイパーパラメータ調整（付録 D）の開発データは，評価データのサブセット SE07 を使用した．WSD タスクの評価データ・方法は，標準評価プロトコル [21] に従った．

表 1 に WSD タスクの性能を示す．提案手法 $\text{SS-WSD}_{\text{emb,kb}}$ は，全事例合計（All 列）において知識ベースの先行研究を上回った．特に TaM 経験則併用時の $\text{SS-WSD}_{\text{kb}}$ は，文のみの情報を用いるにも関わらず，既存の最高精度である COE を 0.8 ポイント上回り，知識ベース WSD の最高精度を更新した．TaM 経験則の寄与は+2.2 ポイント（$\text{SS-WSD}_{\text{kb}} - \text{SS-WSD}_{\text{emb}}$）だった．

次に，TaM 経験則を使用しない場合に注目する．提案手法による適応の効果は+9.3 ポイント（$\text{SS-WSD}_{\text{emb}} - \text{BERT}$）だった．関連語義対の吸引のみで適応を行う先行研究 SREF の効果は+5.4 ポイント（$\text{SREF}_{\text{emb}} - \text{BERT}$）なので，提案手法は語彙知識の活用効率が高いことが示唆される．また品詞別に効果を調べると，動詞の 10.2 ポイントが最大である．この結果は，動詞の関連語義数・異義数が平均 13.0 および 4.1 個と最多であること（付録表 5），すなわち吸引・反発学習の教師信号が豊富な事実と整合する．

最後に，教師あり WSD の先行研究と比較する．用例文の正解語義を用いて語義埋め込みを計算する Sup-kNN および，正解の語義・用例対を用いて BERT をファインチューニングする BEM に対して，提案手法 $\text{SS-WSD}_{\text{emb}}$ との差はそれぞれ +1.4 と −4.1 ポイントだった．提案手法が教師あり WSD の性能に近づきつつあることは，特筆すべき結果である．

6) 自己学習に必要なのはレンマ・品詞のみであるため．形態素解析・複単語表現解析ずみ平文コーパスでも代用できる．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

表 1　全事例合計（All 列）および，サブセット・品詞別の WSD タスク精度．SS-WSD$_{emb,kb}$ は 5 回試行の平均，標準偏差（括弧内），および先行研究最高精度との有意差（スチューデントの両側 t 検定，$*: p < 0.05$）を報告．太字は各区分の最高精度．下線は開発データの精度．“文書” 列は推論時の文書情報の要否．{BEM, Sup-kNN, SREF$_{kb}$, COE} は原論文から引用．

区分	手法	文書	サブセット別					品詞別				All
			SE2	SE3	SE07	SE13	SE15	名詞	動詞	形容詞	副詞	
教師あり	Sup-kNN [18]	×	76.3	73.2	66.2	71.7	74.1	—	—	—	—	73.5
	BEM [19]	×	79.4	77.4	74.5	79.7	81.3	81.4	68.5	83.0	87.9	79.0
知識ベース TaM 経験則無効	BERT	×	67.8	62.7	54.5	64.5	72.3	67.8	52.3	74.0	77.7	65.6
	SREF$_{emb}$ [6]	×	70.3	68.0	60.4	74.2	77.4	76.3	53.5	75.2	76.3	71.0
	SS-WSD$_{emb}$ [提案手法]	×	**74.6***	**73.0***	**65.0***	**77.0***	**79.9***	**78.2***	**62.5***	**79.7***	**80.5***	**74.9***
			(0.5)	(0.6)	(1.3)	(0.5)	(1.0)	(0.4)	(0.7)	(0.3)	(1.5)	(0.3)
知識ベース TaM 経験則有効	SREF$_{kb}$ [6]	×	72.7	71.5	61.5	76.4	79.5	78.5	56.6	79.0	76.9	73.5
	COE [8]	✓	76.0	74.2	**69.2**	**78.2**	80.9	**80.6**	61.4	80.5	81.8	76.3
	SS-WSD$_{kb}$ [提案手法]	×	**77.7***	**75.9***	66.5	78.0	**81.6**	79.3	**65.7***	**84.9***	**84.2***	**77.1***
			(0.5)	(0.6)	(1.0)	(0.5)	(0.9)	(0.3)	(0.8)	(0.4)	(0.8)	(0.3)

表 2　提案手法の一部無効化による性能変化．アブレーション列は無効化した対象．Δ 列は無効化前との差分．全差分は統計的に有意（Welch の両側 t 検定，$p < 0.05$）．

アブレーション	WSD (All)	Δ
SS-WSD$_{emb}$	74.9	—
– 吸引・反発学習	71.6	-3.3
– 自己学習	70.5	-4.4
– 無関連語義 $\mathcal{S}^U$ との反発	69.9	-5.0
– 異義 $\mathcal{S}^N$ との反発	73.5	-1.4
– 文脈埋め込みの変換関数	71.7	-3.2

表 3　語義対，正解語義・用例対の平均 cosine 類似度．

手法	関連	無関連	異義	正解語義・用例
BERT	0.91	0.77	0.87	0.64
SS-WSD$_{emb}$	0.88	0.64	0.78	0.77

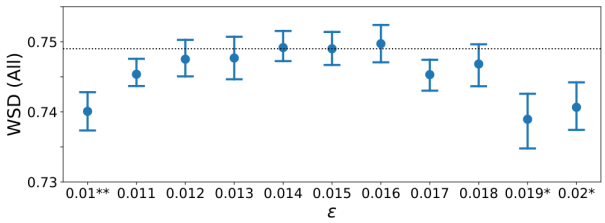

図 2　埋め込み変換関数の距離制約ハイパーパラメータ ϵ（§ 3.2）と WSD 性能の関係．デフォルト設定（$\epsilon = 0.015$）の精度を水平点線で表示．*は精度差分の統計的有意性（Welch の両側 t 検定，$*: p < 0.05$，$**: p < 0.005$）．

5　考察

5.1　目的関数の効果

表 2 に，目的関数（§ 3.3）の一部無効化による性能変化を示す．変化から有効性の源泉を探る．

吸引・反発学習または自己学習を無効化すると，それぞれ 3.3 と 4.4 ポイント性能が低下する．ゆえに両学習の併用は相補的であることが示唆される．

吸引・反発学習から無関連語義または異義との反発を無効化すると，精度はそれぞれ 5.0 および 1.4 ポイント低下する．したがって意味的なつながりがない，または異なる語義を遠ざけることはいずれも有効である．なお無関連語義の寄与が異義を上回る理由は，訓練事例数が考えられる．前者は 255 個[7)] であるが，後者は平均 1.3 個（付録表 5）である．

文脈埋め込みの変換関数を無効化すると，精度は 3.2 ポイント低下する．したがって語義・文脈埋め込みはいずれも適応させるべきであり，SERF のような語義埋め込みの適応のみでは不十分である．

表 3 に類似度の変化を示す．適応により，無関連および異義がお互いに離れ，正解語義・用例が近づいた．この結果は提案手法の狙いと整合している．

7)　(ミニバッチサイズ) – 1 より，256 – 1 = 255 個となる．

5.2　距離制約の効果

図 2 に，変換による移動を制約するハイパーパラメータ ϵ の性能への影響を示す．ϵ に対して精度は逆 U 字曲線を示すことから，厳しい制約（ϵ 小）と緩い制約（ϵ 大）の中間が最適であることが分かる．ゆえに，意味適応を有効に機能させるため，適応時の類似度空間の変化を制限することが重要である．

6　まとめ

本研究では，BERT が計算する語義・文脈埋め込みを，語彙知識を用いて WSD に適応させる手法を提案した．提案手法の性能は，文書情報を用いずに従来の知識ベース WSD 最高精度を上回った．これにより，文単体のみでも意味適応による高精度が実現できることを示した．有効性の主要因は，吸引・反発学習と自己学習の併用，文脈埋め込みの適応，および変換時の距離制約であることを示した．

語彙資源のみで高精度を実現する本手法は，資源が乏しい言語 [22] に好適である．多言語モデルによる英語からのゼロショット転移学習，および多言語語彙資源の学習による多言語 WSD に取り組みたい．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

謝辞

本研究は JSPS 科研費 19H01118 の助成を受けた.

参考文献

[1] Chiraag Sumanth and Diana Inkpen. How much does word sense disambiguation help in sentiment analysis of micropost data? In **Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2015**, pp. 115–121, 2015.

[2] Chihli Hung and Shiuan-Jeng Chen. Word sense disambiguation based sentiment lexicons for sentiment classification. **Knowledge-Based Systems**, Vol. 110, pp. 224–232, 2016.

[3] Zhi Zhong and Hwee Tou Ng. Word sense disambiguation improves information retrieval. In **Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics**, pp. 273–282, 2012.

[4] Niccolò Campolungo, Federico Martelli, Francesco Saina, and Roberto Navigli. Dibimt: A novel benchmark for measuring word sense disambiguation biases in machine translation. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics**, pp. 4331–4352, 2022.

[5] Christiane Fellbaum. **WordNet: An Electronic Lexical Database**. The MIT Press, 1998.

[6] Ming Wang and Yinglin Wang. A synset relation-enhanced framework with a try-again mechanism for word sense disambiguation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 6229–6240, 2020.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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**, pp. 4171–4186, 2019.

[8] Ming Wang, Jianzhang Zhang, and Yinglin Wang. Enhancing the context representation in similarity-based word sense disambiguation. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 8965–8973, 2021.

[9] Ivan Vulic and Nikola Mrksic. Specialising word vectors for lexical entailment. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 1134–1145, 2018.

[10] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In **Proceedings of the 5th Annual International Conference on Systems Documentation, SIGDOC 1986**, pp. 24–26, 1986.

[11] Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of BERT. In **Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019**, pp. 8592–8600, 2019.

[12] Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and José Camacho-Collados. Analysis and evaluation of language models for word sense disambiguation. **Comput. Linguistics**, Vol. 47, No. 2, pp. 387–443, 2021.

[13] William A. Gale, Kenneth Ward Church, and David Yarowsky. One sense per discourse. In **Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, USA, February 23-26, 1992**, 1992.

[14] Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. Hierarchical embeddings for hypernymy detection and directionality. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pp. 233–243, 2017.

[15] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In **IEEE Conference on Computer Vision and Pattern Recognition**, pp. 4690–4699, 2019.

[16] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In **2018 IEEE Conference on Computer Vision and Pattern Recognition**, pp. 5265–5274, 2018.

[17] Ming Wang and Yinglin Wang. Word sense disambiguation: Towards interactive context exploitation from both word and sense perspectives. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing**, pp. 5218–5229, 2021.

[18] Daniel Loureiro and Alípio Jorge. Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation. In **Proceedings of the 57th Conference of the Association for Computational Linguistics**, pp. 5682–5691, 2019.

[19] Terra Blevins and Luke Zettlemoyer. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 1006–1017, 2020.

[20] George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. A semantic concordance. In **Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, USA, March 21-24, 1993**, 1993.

[21] Alessandro Raganato, José Camacho-Collados, and Roberto Navigli. 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**, pp. 99–110, 2017.

[22] Tommaso Pasini. The knowledge acquisition bottleneck problem in multilingual word sense disambiguation. In **Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence**, pp. 4936–4942, 2020.

[23] Yinglin Wang, Ming Wang, and Hamido Fujita. Word sense disambiguation: A comprehensive knowledge exploitation framework. **Knowledge Based System**, Vol. 190, p. 105030, 2020.

[24] Michele Bevilacqua and Roberto Navigli. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 2854–2864, 2020.

[25] Caterina Lacerra, Michele Bevilacqua, Tommaso Pasini, and Roberto Navigli. CSI: A coarse sense inventory for 85% word sense disambiguation. In **The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020**, pp. 8123–8130, 2020.

[26] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In **Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**, pp. 2623–2631, 2019.

[27] Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. XL-WSD: an extra-large and cross-lingual evaluation framework for word sense disambiguation. In **Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021**, pp. 13648–13656, 2021.

[28] Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. Semeval-2007 task-17: English lexical sample, SRL and all words. In **Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval@ACL 2007**, pp. 87–92, 2007.

[29] Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In **Proceedings of the 36th International Conference on Machine Learning**, Vol. 97 of **Proceedings of Machine Learning Research**, pp. 573–582, 2019.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

A 語彙資源

表 4 WordNet 語彙資源の語義の具体例

要素	具体例
語義	`computer%1:06:00::` (sense key)
レンマ	*computer*
Synset	`computer.n.01`
語釈文	*a machine for performing calculations ...*
用例	`Not Available`
同義語	*computer, computing device, data processor, ...*
関連語義	`computing_device%1:06:00::` (同義), `analog_computer%1:06:00::` (下位), ...
異義	`computer%1:18:00::`
無関連語義	`goldfish%1:05:00::`, `chef%1:18:01::`, ...

表 4 に，提案手法が使用する語彙資源の具体例として，語義 `computer%1:06:00::`[8] を示す．

関連語義の定義は先行研究 [6] の論文および実装[9] に従う．具体的には，それぞれの sense key に対して，まず `derivationally_related_forms` 関係にある sense keys を加えて集合を拡張する．次に sense keys が接続する synsets を収集する．そして，各 synset に対して，表 6 で示した意味関係にある synset を加えて集合を拡張する．最後に，synsets に属する sense keys を収集し，関連語義とする[10]．

B BERT による埋め込みの計算

BERT による埋め込みの計算は，先行研究 [23, 24, 6] に倣う．モデルは `bert-large-cased`[11] を用いる．エンコード時には `[CLS]` および `[SEP]` トークンを付与して，次元要素ごとの最終 4 層の和を，サブワードの埋め込みとする．文脈埋め込みは，対象単語を構成するサブワードの平均とする．語義埋め込みの計算は，先行研究 `SREF` [6] に従う．具体的には，所与の sense key に対して，レンマ・同義語 n 個・語釈文・例文 m 個からなる連結文をエンコードする．そして連結文を構成する全サブワードの平均を，語義埋め込みとする．連結文のテンプレートを以下に示す．

```
[レンマ] - [同義語 1], ...,
[同義語 n] - [語釈文] [例文 1] ... [例文 m]
```

具体例として，語義 `computer%1:06:00::` の連結文を示す．

```
computer - computer, computing device, data processor, ... - a machine for performing calculations automatically
```

なお，WordNet 語義の大半は例文がないか，短文である．先行研究 [6] は独自に収集した例文を併用しているが，我々は WordNet に収録された例文のみを使用した．

C Try-again Mechanism (TaM)

Try-again Mechanism (TaM) はいくつかの派生版がある [6, 8, 17]．本研究では簡便性に優れる Wang ら [17] のアルゴリズム[12] を使用する．具体的には，対象単語 w に対する類似度 $\rho_{w,s}$（式 4）が上位 2 個の候補語義 $s \in \{s_1, s_2\}$ について，各語義が属する Coarse Sense Inventory (CSI) [25] の意味カテゴリとの類似度を加算した類似度 $\rho^+_{w,s}$ を計算する．そのうえで $\rho^+_{w,s}$ が最大の語義を選択する．

$$\rho^+_{w,s} = \rho_{w,s} + \max_{s' \in \mathcal{S}^{\mathrm{CSI}}_s} \rho_{w,s'} \tag{14}$$

$\mathcal{S}^{\mathrm{CSI}}_s$ は，語義 s と同じ意味カテゴリに属する語義の集合[13] である．CSI は WordNet 語義を 45 個の意味カテゴリに分類[14] した語義目録である．よって式 14 の右辺第 2 項は，粗粒度で同一視できる語義との類似度を表している．

表 5 WordNet 語彙資源の統計量（関連・異義数は平均値）

要素	名詞	動詞	形容詞	副詞	合計
レンマ数	117,798	11,529	21,479	4,481	155,287
語義数	146,320	25,047	30,002	5,580	206,949
関連語義	7.8	13.0	6.2	3.9	8.1
異義	0.8	4.1	1.2	0.7	1.3

表 6 関連語義の収集に使用する WordNet の意味関係

カテゴリ	意味関係
Sense key	`pertainyms, antonyms`
Synset	`hyponyms, hypernyms, part_holonyms, part_meronyms, member_holonyms, also_sees member_meronyms, entailments, attributes, similar_tos, causes, substance_holonyms, substance_meronyms, usage_domains`

D 実験設定の詳細

訓練時の最適化アルゴリズムは，Adam (学習率 0.001) を用いた．エポック数は 15 とした．提案手法のハイパーパラメータは，ミニバッチサイズ $N_B = 256$，自己学習の重要度 $\alpha = 0.2$，変換時の距離制約 $\epsilon = 0.015$ に設定した．

ハイパーパラメータの調整は，開発データによるハイパーパラメータ最適化を使用した．具体的には，パラメータ空間 $N_B \in \{64, 128, 256, 512, 1024\}$，$\alpha \in [0.1, 10]$，$\epsilon \in [0.001, 0.1]$ を探索[15] したあとで，$\epsilon \in [0.01, 0.02]$ を刻み幅 0.001 でグリッドサーチした．最適化の目標は，TaM 経験則無効時の WSD タスク精度とした．開発データは先行研究 [27] の慣習にならい，評価データのサブセットである SE07 (SemEval-2007 [28]) を使用した．なお最適化の結果，256 より大きいミニバッチサイズは性能に寄与しなかった．また α は ϵ よりも鈍感であった．

WSD タスクの評価データ・方法は，標準評価プロトコル [21] に従った．推論アルゴリズムは，TaM 経験則無効の場合は式 4，有効の場合は式 14 である．評価指標はマイクロ F 値である[16]．訓練は異なるランダムシードで 5 回実行して，平均および標準偏差を報告した．

E 性能が向上しなかった手法

予備実験で性能向上に寄与しなかった手法を報告する．

- 吸引・反発学習で，異義に重み付けすること
- 自己学習で，非最近傍の候補語義を遠ざけること
- 移動距離に対する L2 正則化を併用すること
- 変換関数を可逆 [29] にすること
- 語義・文脈変換関数をパラメータ共有すること

8) 単語 *computer* の “計算機” の意味を指す．なお異義 `computer%1:18:00::` は “計算手” の意味を指す．
9) https://github.com/lwmlyy/SREF
10) 実装には `nltk.corpus.wordnet` package を用いた．
11) 実装には `transformers` package を用いた．
12) https://github.com/lwmlyy/SACE
13) 語義が CSI に未採録の場合は空集合になる．
14) たとえば語義 `computer%1:06:00::` は，意味カテゴリ `CRAFT_ENGINEERING_AND_TECHNOLOGY_` に分類される．
15) 探索アルゴリズムは `optuna` [26] の `TPESampler` を用いた．
16) 提案手法の予測語義数は常に 1 個なので，マイクロ F 値は適合率および再現率と一致する [27].

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).