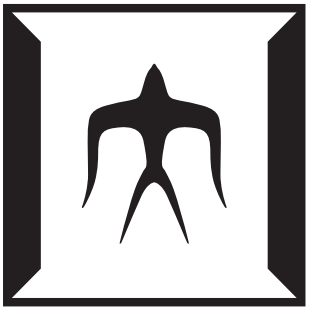
論文 / 著書情報
Article / Book Information

題目(和文)	単語埋め込み表現の語彙知識への意味適応
Title(English)	
著者(和文)	水木栄
Author(English)	Sakae Mizuki
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12629号, 授与年月日:2023年12月31日, 学位の種別:課程博士, 審査員:岡崎 直観,佐久間 淳,篠田 浩一,徳永 健伸,村田 剛志
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12629号, Conferred date:2023/12/31, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

博士論文

単語埋め込み表現の語彙知識への意味適応

水木 栄



東京工業大学 情報理工学院
情報工学系 知能情報コース

本論文は東京工業大学情報理工学院に
博士（工学）授与の要件として提出した博士論文である.

審査委員：
岡崎 直観 教授　（主指導教員）
佐久間 淳 教授　（審査員）
篠田 浩一 教授　（審査員）
徳永 健伸 教授　（審査員）
村田 剛志 教授　（審査員）

Abstract

This study explores the potential of the semantic specialization of word embeddings, leveraging lexical resources to enhance the representation of lexical semantics. It assesses the feasibility of enhancing the performance in recognizing semantic relations between words and identifying the contextually appropriate word senses in a sentence by specializing word embedding—derived from large-scale textual data using deep learning methods—to the structural property of the semantic relations that are stored in lexical resources like WordNet.

Specifically, we propose two conceptually coherent methods. The first method incorporates the characteristics of the hierarchical structure inherent in the concept hierarchy of word meanings, such as transitivity and antisymmetry, into static word embeddings. Subsequently, the second method integrates the characteristics of the semantic network structures, such as semantic relatedness and unrelatedness arising due to the connections between words and word senses, into contextualized word embeddings. We empirically evaluate the effectiveness of the specialized embeddings by applying them to the hypernymy detection task and Word Sense Disambiguation task.

In the first method, we focus on the hypernymy detection between words through the adaptation to the concept hierarchy. This approach transforms static word embeddings into M-ary N-digit hierarchical code representations. Here, hypernyms and hyponyms share the first n digits. This enables the inference of hyponymy relations, aligning with the transitivity and antisymmetry inherent in the hierarchical structure of concept hierarchies. The primary challenge of the proposed method lies in optimizing the function that transforms embeddings into discrete hierarchical codes. This optimization uses gradient de-

scents and hypernym-hyponym word pairs, extracted from the lexical resource, as supervision signals. To address this challenge, we introduce two concepts: the continuous relaxation of code representations and the metric that quantifies the degree of inclusion between code representations as continuous values. Additionally, we incorporate the auxiliary task, which involves reconstructing the input word embeddings from the code representations. We also use random sampling of non-hypernymy but semantically similar word pairs as negative examples. We then applied the degree of inclusion relation among transformed hierarchical codes to resolve the hypernymy detection task suites. As a result, we confirmed that the proposed method outperformed previous methods in the hypernymy classification task. Error analysis indicated that the hierarchical codes effectively capture concept abstractness through the non-zero digit length. However, they occasionally conflate meronymy (whole-part) relations with hyponymy relations, both possessing hierarchical structures. We also examined the characteristics of the hierarchical code assignments to words. We revealed a weak agreement with the structural properties of WordNet's concept hierarchy, such as semantic similarity between words. A key limitation of the first method is its inability to address the context-dependent meanings of polysemous words, stemming from its reliance on static word embeddings. Moreover, it does not leverage knowledge of semantic relations that do not form concept hierarchies.

Conversely, in the second method, we focus on the Word Sense Disambiguation through the adaptation to the semantic networks. This approach transforms the embeddings of the word senses and target words in contexts derived from the contextualized embeddings. This ensures that the similarities between target words and word senses—as well as among word senses themselves—represent their semantic relatedness, like adjacency and neighborhood, within the semantic network. To address the challenges associated with training transformation functions for selecting the semantically closest senses of target words in context, we employ joint optimization of two objectives: the Attract-Repel objective for sense pairs and the self-training objective for target words and word senses. Additionally, we introduce constraints to the transformation functions to limit deviations from the similarities observed in pre-specialized embeddings. We em-

ploy specialized embeddings to disambiguate word senses by choosing the nearest neighboring senses to the target words. Consequently, we confirmed that the proposed method outperformed previous methods in the knowledge-based Word Sense Disambiguation task. We analyzed the similarity characteristics of specialized embeddings and revealed that aligning with the relatedness within the semantic network effectively brings semantically related senses and words closer and sends different senses of the same word or unrelated senses farther away. Furthermore, we observed that the similarities among related, different, and unrelated senses are in sync with the WSD task performance.

Two predominant methodologies represent word meanings: one is word embedding, grounded in statistical methods, and the other is lexical resources, rooted in human knowledge. This research unveils that specializing pre-trained embeddings to the hierarchical and network structures of word meanings in lexical resources contributes to improving the performance in the lexical semantics tasks, such as hypernymy detection and Word Sense Disambiguation. These findings offer insights into semantic specialization of word embeddings and highlight the potential of integrating deep learning-based embeddings with lexical knowledge to achieve better semantic understanding.

Keywords:

representation learning, Semantic Specialization, WordNet, BERT, Word2vec, Hypernymy detection, Word Sense Disambiguation

謝辞

博士後期課程入学爾来はや 6 年が経過しようとする中，研究活動を遂行して学位論文の執筆に至るまでには多くの紆余曲折を経験しました．この間，多くの方々からご指導およびご支援を賜りましたことを心よりお礼申し上げます．

主指導教員である岡崎直観教授には，研究計画の立案から論文の執筆まで，全方位にわたり丁寧かつ有益なご指導を賜りました．先生との議論が研究計画を見直す指針となり，隘路を脱して成果に至る道を見出したことは一度や二度ではありません (水木 2023)．何よりもコロナ禍によって研究活動の継続が危機に陥った状況から立ち直れたのは，紛れもなく先生との建設的な関係性および，それが下支えした率直な対話によるものです．先生の薫陶を受ける機会が得られたことに，心から感謝と誇りの念を抱いております．今後も交流させていただく機会があることを願っております．

本論文の審査員の先生方から賜りました指導にお礼申し上げます．徳永健伸教授には，研究の動機を埋め込みと語彙知識の統合という観点から整理するための助言，ならびに論文草稿への丁寧なレビューを頂戴しました．村田剛志教授および篠田浩一教授には，提案手法の目的，評価タスク，実験設定を明解にするための助言および，グラフ埋め込みや few-shot 学習との関連性の指摘を頂戴しました．佐久間淳教授には，階層コード表現における包含関係の計量や吸引・反発学習の定式化など，提案手法における新規性や計算方法の妥当性に関する指摘や確認を頂戴しました．

支援員の小西由希子さん，雲財祐子さん，古谷奈緒子さんから賜りました数々の事務支援にお礼申し上げます．書籍や研究機材の購入はもとより，出張，英文校正などの研究活動，ひいては博士課程における学務部との手続きなど，経費や事務の処理において大変お世話になりました．

金子正弘特別研究員から頂戴しました研究活動に関する助言にお礼申し上げ

ます．金子さんの研究活動における生産性は驚異的であり，それを可能としている仮説立案や実験設計に関する方法論は，研究者として豊かなキャリアを形成する上できわめて有益なものと考えます．また研究機会の獲得や挑戦を第一とする冒険心は尊敬に値するものであり，一部だけでも見習いたいと感じています．

修了生を含め，博士課程を共にした岡崎研究室在籍者の皆様との交流にお礼申し上げます．2021 年度修了生である平岡達也さんとは，提案手法の改善や実験設計の妥当性など，研究の核心部分について多くの議論をさせていただきました．本研究で取り組んだ語義曖昧性解消は，当初は概念階層への適応による解決を試みるも隘路に陥っていました．階層構造への固執を捨てて意味ネットワークの構造に着目するという方針転換は，平岡さんとの議論が大きな役割を果たしています．2022 年度修了生である丹羽彩奈さんには，自ら体現する形で，対外活動の重要性を啓蒙していただきました．研究コミュニティにおける丹羽さんの貢献は，私自身が NLP D の会への参加や NLP コロキウムでの発表という形で恩恵を被るだけでなく，内向き志向の私に相互交流の意義を啓発してくださるものでした．おなじ 2023 年度に修了を迎える Ao Liu さん，および飯田大貴さんには，博士論文の執筆や発表についての相互レビューや議論などで大変お世話になりました．多くの時間と体力を要する論文審査の過程をともに歩む仲間がいたことは本当に心強く，おふたりの専心を目にしたことは大いに励みとなりました．Marco Cognetta さんには，英文校正で何度も助けていただきました．「英文がよく書けている」という査読者の評価は，あなたのおかげです．Youmi Ma さんには，研究室のホームページ運営などでいつもお世話になりました．楊之申さん，村岡雅康さん，文翔煥さん，Erick Mendieta Molina さん，Vijay Daultani さん，吉川和さん，王安さん，小池隆斗さんなど，皆様とは研究発表会や論文輪読会で有益な議論をさせていただき，また私の研究発表に対しても忌憚のないご意見やご提案をいただきました．研究室内での発表の機会は，知識の習得や意見の交換という役割だけでなく，怠惰な私にとって研究を着実に進捗させる強い動機付けでもありました．それも偏に，立場や年齢差を気にすることなく交流してくださった皆様のおかげです．在籍期間の長さもあり，すべての方々の名前を列挙しない無礼をご容赦ください．皆様が今後もすばらしい成果を成し遂げられることを信じて疑いません．

勤務先である株式会社ホットリンクの同僚および上司の皆様から頂戴したご理解とご支援にお礼申し上げます．上司である山本さんと榊さんは，博士課程で

の知識と経験が組織に好ましい影響をもたらすという強い信念のもと，私の活動を一度たりとも批判することなく常に応援してくださいました．社会人博士という平坦でない道のりを歩めたのは，皆様のおかげです．そして大知さん，多くは語りませんが，あなたの助言はまさしく金言でした．本当に感謝しています．

最後に，常に私の健康を気遣ってくれる両親および祖母に深く感謝します．30 代での挑戦をためらう私を後押しし，応援してくださいました．自然言語処理技術はいよいよ社会に大きな影響を及ぼすまでになり，きっと皆さんがその影響を身近に感じる機会も増えてゆくことでしょう．願わくは，皆さんを含む社会全体への助けとなるような形で，私なりの方法で恩返しをしていきたいと思っています．

目次

1 序論 **1**

1.1 単語の意味 1
1.2 単語埋め込みと語彙資源 3
1.3 意味適応による統合 7
1.4 本研究の方策 10
1.4.1 概念階層への適応による上位下位関係識別 10
1.4.2 意味ネットワークへの適応による語義曖昧性解消 12
1.5 貢献 13
1.6 論文の構成 14

2 関連研究 **16**

2.1 静的埋め込み 17
2.1.1 概要 17
2.1.2 学習方法 18
2.1.3 意味表現としての特徴 20
2.2 文脈依存埋め込み 22
2.2.1 概要 22
2.2.2 学習方法 23
2.2.3 意味表現としての特徴 27
2.3 語彙資源 29
2.3.1 WordNet 29
概要 29
構成および意味関係 29
意味の階層構造 32

意味のネットワーク構造 33
統計量 34
2.3.2 WordNet に関連する言語資源 35
BLESS 36
HyperLex 37
Unified Evaluation Framework for WSD 37
WSD Hard Benchmark 38
SemCor 39
Coarse Sense Inventory 39
BabelNet 40
分類語彙表 40
2.4 単語埋め込みと語彙資源の統合 42
2.4.1 静的埋め込みと語彙資源の統合 42
2.4.2 文脈依存埋め込みと語彙資源の統合 45

3 概念階層への適応による上位下位関係識別 48
3.1 概要 48
3.2 関連研究 51
3.2.1 Order Embeddings 51
3.2.2 意味適応 52
3.2.3 コード表現 52
3.3 提案手法 53
3.3.1 階層コードおよび上位下位関係の定義 54
3.3.2 変換器 54
3.3.3 上位下位関係の計量 56
3.3.4 目的関数 60
3.3.5 非上位下位語ペアの生成 62
3.4 実験結果 63
3.4.1 評価タスク・データセット・推論方法 63
3.4.2 学習方法 64
単語埋め込みおよび語彙資源 64
最適化およびハイパーパラメータ 66

3.4.3 結果 66
3.5 分析 67
3.5.1 分類タスクの誤り分析 67
3.5.2 ランキングタスクの誤り分析 68
3.5.3 有効性の要因 69
3.5.4 階層コードの割り当て特性の分析 70
分析方法 71
WordNet の概念階層との比較 72
基数および桁数の影響 74
3.5.5 間接的な上位下位語ペアの影響 76
3.5.6 非上位下位語ペア生成の影響 77
3.5.7 多義語における上位下位関係の分析 79
分析方法 79
具体例の観察 79
因子との相関分析 80
3.6 本章のまとめ 82

4 意味ネットワークへの適応による語義曖昧性解消 85
4.1 概要 85
4.2 関連研究 87
4.2.1 知識ベース語義曖昧性解消 87
4.2.2 教師あり語義曖昧性解消 89
4.2.3 埋め込みの意味適応 89
4.3 提案手法 90
4.3.1 BERT による埋め込みの計算 91
4.3.2 語義埋め込みの計算 92
4.3.3 変換関数 93
4.3.4 訓練の目的関数 94
吸引・反発学習 94
自己学習 96
4.3.5 Try-again Mechanism (TaM) 経験則 97
4.4 実験設定 98

4.4.1 訓練 98
4.4.2 評価 99
4.4.3 ベースライン 99
4.5 実験結果 100
4.6 分析 102
4.6.1 BERT 埋め込みの特性 102
4.6.2 目的関数の効果 103
4.6.3 距離制約の効果 105
4.6.4 自己学習訓練データ量の影響 105
4.6.5 類似度の変化 106
4.6.6 非一般的な語義に対する有効性の分析 108
評価 109
ベースライン 109
実験結果 110
4.7 本章のまとめ 112

5 結論 **115**

参考文献 **121**

第 1 章

序論

1.1 単語の意味

本研究では，大規模言語データから学習した単語埋め込みを，人間が構築した語彙資源の知識に適応させることで，単語間の意味的関係を識別したり，多義語の意味をひとつに特定する性能の改善に取り組む．

単語の意味を取り扱う方法は，自然言語処理の中核的課題である．単語は文ひいては文章を構成する根源的な要素であるため，単語間の意味的関係を識別したり，文内で単語が表す意味（語義）を特定する技術は，コンピュータで自然言語を解析するさまざまな場面で必要とされる．

単語間の意味的関係を識別する問題で特に重要なのは，文とは独立に与えられた単語が，具象 $\rightarrow$ 抽象の関係である上位下位関係であるか否かを識別する技術である．単語対が上位下位関係か否かを識別する問題は，上位下位関係識別と呼ばれている．

- 犬に対する動物は，下位語と上位語の関係である
- 犬に対する鳥は，下位語と上位語の関係ではない

上位下位関係識別の重要性を示す例は，テキスト間の論理関係推論 (Dagan et al. 2013) である．

(1) 彼は犬を飼っている．

(2) 彼は動物を飼っている．

(3) 彼は鳥を飼っている．

例文に含まれる犬 → 動物は上位下位関係だが，犬 → 鳥はそうではない．上位下位関係は，下位概念が上位概念の特徴や性質を特化する形で継承することから生じる概念階層に由来する．したがって (1) が真ならば (2) も真であると推論できるが，(3) が真かは推論できない．また上位語への言い換えは概念の抽象化にあたるため，言い換えや文書要約などとも関連性が深い．

単語はしばしば複数の語義に対応する．このため，文内で単語が表す語義を周辺の単語（すなわち文脈）をふまえて正しい語義を特定することは，おなじく重要な技術である．テキスト内におかれた対象語の意味を候補語義の中から選ぶ問題は，語義曖昧性解消（WSD: Word Sense Disambiguation）と呼ばれている．

- *We need to justify the margins.*

(a) “正当化する” の意味

(b) “余白を調節する” の意味

(c) “神が罪人を赦す” の意味

この例文における単語 *justify* は “余白を調節する” の意味で用いられているので，正解は (b) である．

語義曖昧性解消が貢献する例は，正確な翻訳 (Campolungo et al. 2022) である．

- *We need to justify the margins.*

(1) 余白を正当化する必要がある．

(2) 余白を揃える必要がある．

justify の語義が (b) であると特定できていれば，正しい訳文は (1) ではなく (2) であることが判断できる．

上位下位関係識別および語義曖昧性解消は，前述したテキスト間論理関係推論や機械翻訳のほかにも，評判分析 (Sumanth and Inkpen 2015, Hung and Chen

2016)，テキスト生成 (Biran and McKeown 2013)，情報検索 (Zhong and Ng 2012) での有効性が指摘されている．このため，自然言語処理研究における語彙意味論や意味解析の分野で長く取り組まれてきた．そして 2020 年代に入って，深層学習による生成型 AI が汎用的な言語処理の方法として普及する中でも依然として有意義である．なぜならば，AI の応答が正しいかどうか，生成されたテキストを検証するために必要とされるからだ．単語とその意味は膨大かつ多様であるため，AI が知らない単語や間違えて記憶している語義が存在しないことは保証できない．著名な百科事典である Wikipedia 英語版の見出し語数は約 700 万語に至るし，辞書様態の語彙資源である Princeton WordNet (Fellbaum 1998) に採録された語義数は 20 万件を上回る．

単語間の意味関係を識別したり，複数の語義から文脈に合致するひとつを選択するには，単語の意味を機械で取り扱えるような表現にする技術が必要である．それでは，単語の意味を表現するにはどのような方法論が用いられているだろうか．

1.2 単語埋め込みと語彙資源

単語の意味を表現する方法論は，統計に基づく方法論，すなわち単語埋め込みと，人間の知識に基づく方法論，すなわち語彙資源に大別される．本研究では，これらを統合的に利用する方法を模索する．そこで本節では，両者の特徴ならびに長所と短所を紹介したうえで，統合がもたらす意義を述べる．

統計に基づく方法論は，2010 年代以降の深層学習を用いた自然言語処理における主流である．具体的には，単語を数百から数千次元のベクトルで表現し，大規模言語データを使ってベクトルを計算する．計算したベクトルのことを単語埋め込み，ベクトルを計算することを（分散）表現学習とよぶ．単語埋め込みの特徴は，分布仮説に基づき，文脈類似性をベクトルの近さで表すことである．分布仮説では，単語の意味は周辺の単語，すなわち文脈によって決まると仮定する．したがって，出現する文脈が似ている単語は互いに似たベクトルを割り当てる．この特徴は，スケーラビリティという長所と，詳細な意味知識の欠如および解釈性の不足という短所をもたらす．分布仮説に基づく学習はしばしば，テキスト内の空欄にあてはまる単語を予測するタスクによって行われる．このタスクはテキストに空欄を作ることで機械的に問題を生成できるため，大規模言語データ

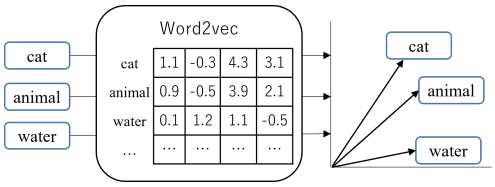


図 1.1: 静的埋め込み

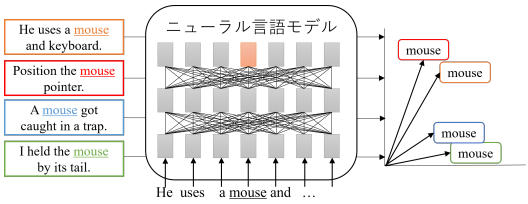


図 1.2: 文脈依存埋め込み

を用いて学習することで，数百万もの語彙を網羅できる．一方で穴埋め問題を解く手がかりは周辺の単語だけであることから，自ずと類似した文脈には似たような単語が予測される．その結果，対義語 (*tasty, tasteless*) や同位語 (*spring, summar*) のように，意味的に関連するが必ずしも類義ではない単語のベクトルも互いに近くなる (Camacho-Collados and Pilehvar 2018)．すなわち，ベクトルの近さは必ずしも意味関係の違いを捉えるとは限らない．そもそも単語間の関係性がベクトルの類似度のみで表されるため，近さ・遠さでは定義不能な上位下位関係や対義関係は表現しようがない．また語義に立脚した学習ではないため，単語埋め込みを特定の意味に紐づけて解釈することはできない．

単語埋め込みの計算方法は，文脈依存性の有無によりふたつに大別される．文脈依存性を考慮しない方法は静的埋め込みと呼称され，文内の共起語を教師信号として単語ごとにひとつのベクトルを割り当てる（図 1.1）．このため，静的埋め込みは文脈から独立した意味の取り扱いに適している．文脈依存性を考慮する方法は文脈依存埋め込みと呼称され，単語予測タスクであらかじめ訓練されたニューラル言語モデルに文を入力し，その隠れ状態ベクトルを取り出して単語埋め込みとみなす（図 1.2）．したがって文脈依存埋め込みでは同じ単語でも文が異なれば違うベクトルになり，これにより文脈に依存した意味の取り扱いが可能になる．

人間の知識に基づく方法論は，1970 年代に発展した合理主義・形式主義による自然言語処理における中心的なアプローチである．具体的には，単語の意味を離散的な記号として扱い，辞書やシソーラス（類義語辞典），オントロジーとして体系化する．これらは語彙資源とよばれる．語彙資源における単語の意味は，語義を演繹的に示す語釈文や，具体例から帰納的に示す用例文により表現される．また単語間の意味関係は，ふたつの単語および意味関係タイプからなる三つ組みにより表現される．たとえば著名な語彙資源である WordNet（図 1.3）で

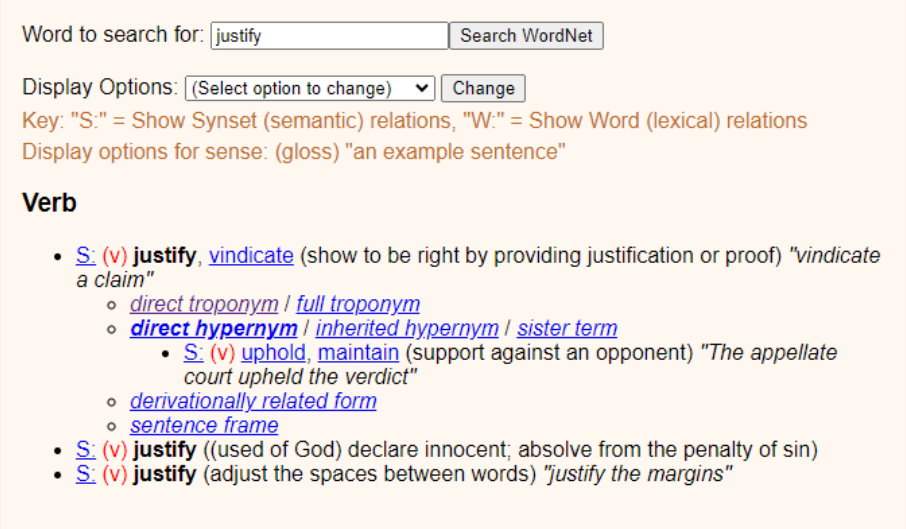


図 1.3: WordNet オンライン版で *justify* を検索して “正当化する” の上位概念を表示した画面.

は単語 *justify* における “正当化する” の意味が以下のように記述されている.

- 見出し語: *justify, vindicate*
- 語釈文： *show to be right by providing justification or proof.*
- 用例文： *vindicate a claim.*
- 意味関係： *vindicate* (同義), *uphold* (上位), *excuse* (下位), ...

語彙資源の特徴は，人間の知識に基づき，人間が理解可能な自然言語で意味を表すことである．この特徴は，緻密さおよび解釈可能性という長所と，スケーラビリティの欠如および定量的推論の困難さをもたらす．たとえば語義を追加すれば単語の意味を細分化できるし，語義間に意味関係を定義すれば意味どうしの関連性を表現できる．また単語の意味は語釈文や用例文を読めば理解できるし，人間が編集することもできる．一方で語彙資源の構築には知識・労働集約的な作業が伴うため，実用上十分といえるほどに大規模化するには時間と人手を要する．また，ある文の単語がどの意味，つまり特定の用例がどの語釈文に該当するかをコンピュータで計算する方法は自明ではなく，語彙資源の利用者が考える必要がある．

以上の説明から，単語埋め込みと語彙資源の統合による利点をふたつ指摘できる．ひとつめは，長所の相互補完である．単語埋め込みのスケーラビリティと語彙資源の緻密さを組み合わせれば，数百万もの語彙を網羅しつつ，意味の違いや単語間の詳細な意味関係を捉えられるだろう．ふたつめは，深層学習と人間に共通の作業基盤をもたらすことである (d'Avila Garcez and Lamb 2020, Maruyama 2021)．たとえば単語埋め込みを語釈文に紐づけられるならば，特定のベクトルがどんな意味を表現しているのか解釈でき，深層学習モデルの振る舞いを理解することに寄与するだろう．通時的な新語の産出や意味の変化に対しても，見出し語の追加や語釈文の更新により追従できる．ほかにも，語彙資源と整合するように埋め込みを更新する，いわば表現学習の誤りを人間の知識で修正する使い方も考えられる．一般的でない単語や語義，表現学習用のテキストが不足する少数言語や特定ドメインでは，語彙資源による補完が有効かもしれない．

これらの利点は，本研究の目標である上位下位関係識別および語義曖昧性解消の性能改善という工学的な観点からも魅力的である．相互補完性は上位下位識別の精度改善に寄与しうるし，解釈可能性は語義曖昧性解消に応用できるためだ．一方で，単語埋め込みはベクトル，語彙資源は語義や自然言語という記号を用いており，表現形式が異なる両者を組み合わせて利用する方法は自明ではない．

教師あり学習はどうだろうか．すなわち，単語埋め込みを特徴量，語彙資源から得られる知識を教師信号として，特定のタスクを解くモデルを構築する方策である．たとえば上位下位関係識別の場合は，静的埋め込みを特徴量，上位下位語対を教師信号として，上位下位関係か否かの識別器を訓練できる．また語義曖昧性解消の場合は，対象単語の文脈依存埋め込みを特徴量，単語の正解語義を教師信号として，語義識別器を訓練できる．ただし後者については語彙資源の用例文では規模が不十分なので，単語ひとつひとつに正解語義を注釈した用例文を集めた SemCor コーパス (Miller et al. 1993) のような専用の言語資源を用いる必要がある．いずれにせよ，教師あり学習はふたつの短所が知られている．ひとつめは，汎化性能である．たとえば上位下位関係識別では，単語埋め込みを特徴量とする識別器は単語間の関係ではなく各単語の上位語らしさを記憶する傾向があり，未知語を対とする場合や，上位語と下位語を入れ替えた場合の精度が低いという Lexical Memorization (Levy et al. 2015) の問題が報告されている．同様に語義曖昧性解消においては，語義使用頻度のロングテール性に起因する

汎化性能の問題が知られている．具体的には，語義注釈付き用例文コーパスの事例はよく使われる語義に偏っている（Knowledge acquisition bottleneck: Pasini (2020)）という問題があるため，コーパスに採録されない，または非典型的な語義は識別精度が低いことが報告されている (Maru et al. 2022)．たとえば *chair* には“椅子”または“教授職”の語義があるが，前述した SemCor における後者の使用率は 10%未満である．ふたつめは，語義注釈付き用例文の作成コストである．語釈文を参照しながら単語の語義を区別して付与する作業は専門性が高く費用と時間を要する (Bevilacqua et al. 2021) ことから，コーパス構築は容易ではない．これは多言語対応や実用化を阻害する要因になる．

教師あり学習によって単語埋め込みから情報を取り出す方法を学ぶ方策は短所があるとわかった．それでは，語彙資源から得られる知識を単語埋め込みに注入することはできないだろうか．そのような背景で本研究が注目するのが，意味適応（Semantic Specialization: Nguyen et al. (2017)）という方策である．

1.3 意味適応による統合

意味適応は，語彙資源から得られる意味に関する知識を教師信号として，計算ずみの単語埋め込みを変換または更新する方策である．これにより，特定のタスクを解く識別器を訓練することなく，変換・更新した単語埋め込みの類似度などを用いて直接問題を解く．わかりやすい例は，意味的な関連性を類似度に反映するものである． Faruqui et al. (2015) は，上位下位語対や同義語対を互いに近づけるように埋め込みを更新すると，意味的に類似する・トピック的に関連する単語を埋め込み類似度で見つける精度が高くなることを示した．

上位下位関係識別では，意味適応の手法が高い性能を発揮することが報告されている．たとえば Vulic and Mrksic (2018) は，上位下位語対を教師信号として，上位語のベクトルを短く，下位語のベクトルを長くするとともに，両者のベクトルがなす角を小さくする方法を提案した．すなわち，語義の抽象度をノルムに，上位下位関係の有無を cosine 類似度に反映することで，任意の単語対に対してノルムの差と cosine 類似度を用いて上位下位関係か否かを推論する方法を提案した．

語義曖昧性解消では，意味適応によって語彙資源のみで問題を解くことにより，実用化の障壁を下げる方法が提案されている．たとえば Wang and Wang

(2021) は，語釈文などを教師信号として，対象語と同じ空間上で語義の埋め込みを計算する方法を提案した．具体的には，見出し語・語釈文・用例文の連結を文とみなして，文を構成する単語の文脈依存埋め込みについて平均を取る．そのうえで対象語と語義の埋め込みを用いて，最近傍法により正解語義を予測する．すなわち語義を特定したい単語を対象語として，その文脈依存埋め込みとの cosine 類似度が最大となる語義を選択する．本手法は教師あり学習による手法の性能にはおよばないものの，語義注釈付き用例文コーパスへの依存性を解消することに成功している．また Wang and Wang (2021) は，意味関係的知識の有効性も報告している．具体的には，上位下位や全体部分などの意味関係でつながる語義どうしの埋め込みを加重平均すると，対象語と正解語義が近づき，最近傍法の性能が改善することを示した．

ここまで述べたように，意味適応に基づく既存手法は，上位下位関係識別では一定の性能を実現している．また語義曖昧性解消では，教師あり学習の性能には劣るが，実用化しやすい方法論となっている．一方で意味適応に基づく手法は，既存研究では取り組まれていない問いが残されている．それは，

- 意味関係の背景にあるデータ構造へ適応させることで，さらに性能を改善できるか．

という疑問である．すなわち，上位下位語対や対義語対など，語彙資源に明記されている意味の二項関係にのみ適応するのではなく，それらを生じる元となった意味の階層構造やネットワーク構造の特徴に対して適応させることで，上位下位関係識別や語義曖昧性解消の性能を改善できるか，という問いである．

まず上位下位関係識別について考えてみよう．上位下位関係は，上位概念の特徴や性質を下位概念が継承する，抽象から具体に至る意味の概念階層に起因している（図 1.4）．このため上位下位には順序または包含関係があり，実際に上位下位語対は推移律および反対称律を満たす．たとえば下位語 $\rightarrow$ 上位語である *cat*$\rightarrow$*carnivore* および *carnivore*$\rightarrow$*animal* に推移律を適用すると *cat*$\rightarrow$*animal* の上位下位関係が導かれ，反対称律を適用すると *carnivore*$\rightarrow$*cat* が上位下位関係でないことが導かれる．このため上位下位関係識別タスクでは，推移律と反対称律が有効な帰納バイアスかもしれないこと，および応用上の有用性が指摘されている (Camacho-Collados 2017)．たとえば上位下位関係識別の応用であるタクソノミの自動構築（Automatic Taxonomy Construction: Bordea et al. (2016, 2015)）

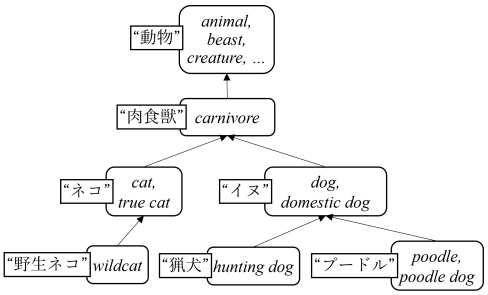


図 1.4: WordNet の上位下位関係が形成する意味の概念階層．わかりやすさのため一部の語義や単語を省略．

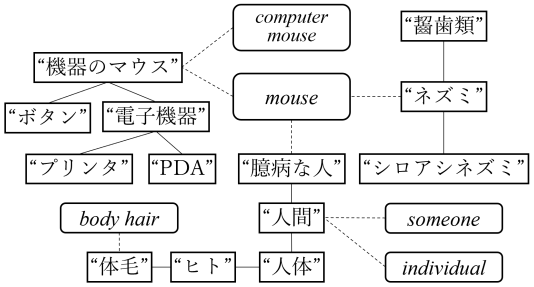


図 1.5: WordNet の単語および語義が形成する意味ネットワーク．明朝体は語義，斜体は単語を表す．破線および実線は，それぞれ単語の意味および意味関係によるつながり．

では，識別した上位下位語対の集合を閉路などの矛盾がない階層構造に組み立てる必要がある．したがって，もしもはじめから推移律と反対称律に沿う推論がなされれば，階層に組み立てる際に互いに矛盾が生じないため望ましいといえる．これに対して単語埋め込みの主流であるベクトル空間での cosine 類似度は，順番を反転させても値が変わらないので反対称律を満たさない．また前述したとおり，ベクトルの長さと向きでそれぞれ概念の抽象度と意味的な関連度を表現し，両者を併用して上位下位関係らしさを定量化する既存研究 (Vulic and Mrksic 2018) も存在するが，あくまでベクトルで表現できる範囲の工夫にとどまっている．したがって，順序あるいは包含関係を定義可能な表現を用いて推移律と反対称律に沿った推論を促すことで，結果的に上位下位関係識別の精度を改善する余地があると考えられる．なお順序または包含関係を表現できる埋め込みは Order Embeddings (Vendrov et al. 2016) という名称で研究されており，その中には意味の概念階層を埋め込む研究も存在する (Athiwaratkun and Wilson 2018)．しかし既存手法は上位下位語対を教師信号として直接に単語埋め込みを学習するものであり，学習ずみ単語埋め込みと語彙資源を統合するものではない．このため，大規模言語データから学習できる文脈類似性の情報が活用されておらず，また語彙資源に存在しない単語は扱えないという欠点がある．

次に語義曖昧性解消について考えてみよう．単語および語義は，単語が取りうる語義および，語義間の意味関係によって結ばれており，単語と語義をノードとするネットワークを構成する（図 1.5）．このため単語と語義または語義ど

うしは，ネットワークの構造において特徴的な関係を持つ．たとえば単語とその語義（*mouse*—“ネズミ”）または，意味的に関連する語義（“ネズミ”—“齧歯類”）は隣接している．同じ単語の異なる語義（“ネズミ”—*mouse*—“機器のマウス”）は単語の近傍（単語に隣接するノードの集合）に属しており，単語を介して 2 ホップで結ばれている．意味的な関連性の低い語義（“臆病な人”, “体毛”）はいずれにも該当しない．このようなネットワーク上のノードとエッジの配置における特徴を埋め込み間の類似度に反映することで，類似度が単語および語義の意味的な近さ・遠さを表すようになり，結果的に対象語と正解語義の埋め込みを近づけられる可能性がある．これに対して，関連語義どうしを近づける既存手法 (Wang and Wang 2020) は，隣接する単語と語義，および隣接しない語義の情報が使われていない．換言すると，意味ネットワークから取得する教師信号を増やす余地がある．ただし多義語においては，単語と語義を近づけてよいのは文脈が合致する場合のみであり，無差別に近づけてよいわけではない．たとえばある文における *mouse* の埋め込みを “ネズミ” や “機器のマウス” と近付ける・遠ざけることをどうやって決めたらよいだろうか．そもそも文脈と語義の一致を判断すること自体が語義曖昧性解消の目的であり，単語と語義の隣接関係をどのように埋め込み間の類似度として反映すればよいかは自明ではない．

意味適応にもとづく手法は，意味関係の背景にあるデータ構造の特徴へ適応させるというコンセプトを導入することで，既存研究の課題を解決あるいは改善できる可能性があることを述べた．本研究ではその具体的な方法論を提案し，上位下位関係識別および語義曖昧性解消における有効性を実験的に検証する．

1.4 本研究の方策

本研究では，意味適応の方策，とりわけ意味関係の構造的特徴へ適応させるという一貫したコンセプトのもとで，単語埋め込みと語彙資源を統合する手法をふたつ提案する．各手法の概要を，表 1.1に示す．

1.4.1 概念階層への適応による上位下位関係識別

ひとつめの手法は，単語埋め込みを意味の概念階層に適応させて，上位下位関係識別を解くものである．具体的には，上位下位語対を教師信号として，静的単語埋め込みを階層コード表現に変換するモデルを学習することで，上位下位

表 1.1: 提案手法の概要

タスク	上位下位関係識別	語義曖昧性解消
データ構造	概念の階層構造	単語と語義のネットワーク構造
用いる語彙知識	上位下位語対	見出し語，語釈文，用例文，意味関係
用いる埋め込み	静的単語埋め込み	文脈依存埋め込みにもとづく 対象語および語義の埋め込み
新規性	単語を階層コードで表現	単語と語義および語義どうしの 近さ・遠さを変更
解決方策	階層コードのペアに対する 微分可能な包含関係計量	語義どうしの吸引・反発学習および 単語と語義の自己学習
貢献	上位下位関係識別の性能改善	知識ベース語義曖昧性解消の最高精度

関係を階層コード間の包含関係によって推論する手法である．階層コードとは，M 進 N 桁，ただしひとたび 0 が出現したら後続桁はすべて 0 となる離散ベクトルであり，包含関係を定義可能である．シラバスにある授業コードの先頭桁を見れば受講対象者の上位概念である学年や学部などがわかるように，単語の階層コードでは上位語の非ゼロ桁が下位語の先頭桁と一致することをめざす．たとえば *animal* は *cat* の上位語なので，*animal* が $(3, 7, 0, 0)$，*cat* が $(3, 7, 1, 2)$ という具合である．深さ M，最大幅 N の木構造は，幅優先符号化により M 進 N 桁の階層コードに変換できる．これが示唆するように，階層コード表現への変換を直感的に解釈すると，単語を階層構造のノードに割り当てることで，推移律および反対称律を満たす推論を可能にする発想である．課題は，このような都合の良い（*animal* のコードが *dog* や *cat* のコードを包含するような）変換をするようにモデルを最適化することである．訓練データとして与えられた単語ペア，たとえば *mouse* と *animal* をそれぞれモデルで階層コードに変換しても，離散的なコードどうしの包含関係は「する」か「しない」かの二値である．よって包含するようにモデルパラメータを更新したくても勾配が計算できない．そこで本研究では，連続緩和 (Maddison et al. 2017) したコード表現を用いること，およびコード間の包含関係を 0 から 1 の連続で微分可能な値として計量する方法を考案した．連続緩和したコードは，数学的にはコードの確率分布を決める分布パラメータの役割を担う．つまり提案手法は，単語埋め込みをコードではなくコードの確率分布に変換し，包含関係になるコード対が出力される期待値を計算することで，包含関係を 0 から 1 で計量することを可能にしている．提案手法で学習したモデルを使うと，単語埋め込みをコード表現に変換できる．コードどうしの包含関係の値をもとに上位下位関係を推論すると，上位下位関係識別タスクの一部で

既存手法を上回ったことを報告する．

本手法の限界は，単語の意味をひとつに限定することである．上位下位関係識別は文脈非依存の単語の意味を問うタスクなのでむしろ好都合だが，語義曖昧性解消では意味の文脈依存性を扱う必要がある．また階層構造を持つ意味関係は名詞や動詞に特有であり形容詞や副詞には存在しないうえに，さまざまな意味関係の一部にすぎない．そこでふたつめの手法は，文脈依存埋め込みおよび，階層性の有無にこだわらずすべての意味関係を用いるように発展させる．これにより，文脈によって変化する単語の意味を扱えるようにするとともに，活用する意味関係的知識の幅を広げる．

1.4.2　意味ネットワークへの適応による語義曖昧性解消

ふたつめの手法は，文脈依存埋め込みから計算した対象語および語義の埋め込みを意味ネットワークの構造に適応させて，知識ベースアプローチにより語義曖昧性解消（WSD）を解くものである．知識ベース WSD の有望な方法論は，埋め込みを用いた最近傍法，すなわち対象語の埋め込みにもっとも近い語義埋め込みを選択する方法である (Wang and Wang 2020)．最近傍法の核心は，語釈文と用例の対応付けである．すなわち語義注釈付き用例文を使用せず，語彙資源のみを用いて，埋め込み空間上で対象語と正解語義を近づけることができれば，性能向上の可能性がある．そこで本研究では，意味ネットワーク上の単語および語義の結びつきに対する特徴への適応を通じて，意味的な関連性を埋め込み間の類似度に反映する手法を提案する．具体的には，吸引・反発学習および自己学習を併用して，埋め込みの変換により類似度を変更する関数を学習する．まず吸引・反発学習では，ネットワーク上で隣接する語義を互いに近付け，単語の近傍語義どうしおよび隣接でも近傍でもない語義を遠ざける．これは，意味的に関連する語義を近付け，意味的に関連しない語義および，同じ単語の異なる語義を遠ざける意図である．つぎに自己学習では，平文コーパスを訓練データとして，対象語に隣接する語義との類似度を変更する．ただし対象語が複数の語義と隣接する多義語の場合は，文脈に合致する語義のみと近付けたい．たとえば *mouse* は{“ネズミ”, “機器のマウス”, “臆病な人”}と隣接するので，文内の周辺単語をふまえて近そうな意味を選んで近付けたい．そこで自己学習では，対象語の埋め込みに最も近い語義と近付ける．これは最近傍法で予測した語義を擬似正解とみ

なすブートストラップ法である．つまり自己学習は，文脈をふまえて対象語と意味が似ている語義を近付けることを意図している．既存研究 (Wang and Wang 2021) に対する提案手法の優位性は，吸引・反発学習と自己学習の併用により，語義どうしおよび単語と語義の両方について類似度を変更できること，および意味的な関連性だけでなく，非関連性および同じ単語の異なる意味という情報も活用することである．また適応ずみの埋め込みを用いて対象語の最近傍語義を選択することで，知識ベース WSD の最高精度を達成したことを報告する．

1.5 貢献

本研究の貢献は，深層学習の手法および大規模言語データを用いて学習した単語埋め込みを，人間が構築した語彙資源に含まれる単語の概念階層および意味のつながりがなす構造；階層構造およびネットワーク構造に適応させる手法を提案し，単語間の上位下位関係および，文脈に依存した単語の意味を識別する性能が改善可能だと示したことである．具体的には以下の通りである．

- **単語埋め込みを意味の階層構造に適応させる手法の提案** 静的単語埋め込みを，階層性を備えたコードを連続緩和した表現に変換するモデルアーキテクチャを提案した．また，コードのペアに対して包含関係の程度を連続変数として計量する方法を提案した．これにより，上位下位語対を教師信号として変換モデルを最適化することを可能にした．

- **上位下位関係識別タスクにおける有効性の実証** 階層コードどうしの包含関係によって上位下位関係を識別する実験により，従来の手法を上回る性能を実証した．また，各単語に対する階層コードの割り当てを単語の意味によるクラスタリングの観点から分析し，WordNet が持つ上位下位関係の階層構造と類似した割り当てがなされることを明らかにした．

- **単語埋め込みを意味のネットワーク構造に適応させる手法の提案** 対象語の文脈依存単語埋め込みおよび，語釈文等から計算された語義埋め込みを，意味ネットワーク上での単語および語義の結びつきを類似度に反映するように埋め込みを変換する手法を提案した．具体的には，語義間の意味関係を教師信号とする吸引・反発学習と，対象語の意味に近い語義を自ら選んで

教師信号とする自己学習を併用することで，語彙資源と平文コーパスのみを訓練データとして埋め込みを変換するモデルを学習する方法を提案した．

- **語義曖昧性解消タスクにおける有効性の実証** 適応ずみの埋め込みを用いて，対象語の最近傍語義を選ぶことで語義曖昧性解消タスクを解く実験により，知識ベースアプローチによる従来の手法を上回る性能を実証した．また，適応前後での埋め込み間の類似度の変化を分析し，対象語は文脈が合致する語義に近づき，意味的に関連しない語義，および同じ単語の異なる語義は互いに遠ざかるという想定どおりの変化が生じることを明らかにした．

1.6 論文の構成

本論文の構成は以下の通りである．

2. 準備および関連研究

2 章では，本研究の基礎をなす単語埋め込みおよび語彙資源について説明したのちに，埋め込みに語彙資源の情報を統合する既存研究を紹介する．単語埋め込みについては，静的埋め込みおよび文脈依存埋め込みの 2 種類について，概要，学習方法，および学習ずみの埋め込みが捉える情報を説明する．語彙資源については，WordNet の概要を説明したのちに，意味関係の階層構造およびネットワーク構造としての特徴を述べる．また WordNet と関連する言語資源のほか，上位下位関係識別や語義曖昧性解消タスクで用いられる評価データセットなどを紹介する．埋め込みと語彙資源の統合については，本研究が採用する意味適応による方策のほか，文脈依存埋め込みについては埋め込みの計算過程で語彙資源を活用する方策についても述べる．上位下位関係識別および語義曖昧性解消の各タスクに特化した関連研究は，3 章および 4 章で述べる．

3. 概念階層への適応による上位下位関係識別

3 章では，ひとつめの提案である，静的単語埋め込みを階層コード表現に変換して概念階層に適応させる手法を説明する．次に，得られたコード表現を用いて上位下位識別タスクを解いた結果を報告する．また，提案手法が有効に機能す

る要因の分析結果および，階層コードの割り当て特性と WordNet における意味の概念階層との比較に関する分析結果を報告する．

4. 意味ネットワークへの適応による語義曖昧性解消

4 章では，ふたつめの提案である，文脈依存埋め込みにもとづく対象語および語義埋め込みどうしの距離を変更して意味ネットワークの構造に適応させる手法を説明する．次に，適応ずみ埋め込みを用いた最近傍法によって知識ベース語義曖昧性解消タスクを解いた結果を報告する．また，タスク性能に影響する要因の分析結果および，適応による埋め込み類似度の変化と意味的な関連性との関係および，タスク精度との関係についての分析結果を報告する．

5. 結論

5 章では，大規模言語データから学習した単語埋め込みを，語彙資源から得られる知識へ適応させることの有効性について論じる．最後に，本研究の内容を発展させる際の有望な方向を述べる．

第 2 章

関連研究

本研究では，大規模な言語データから学習した単語埋め込みを，人間が構築した語彙資源の知識に適応させる手法を提案する．本章では，提案手法や実験の説明に必要な準備として，単語埋め込みの技術および語彙資源の内容を解説する．単語埋め込みについては，静的埋め込みおよび文脈依存埋め込みの 2 種類を解説する．語彙資源については，WordNet および関連する言語資源を解説する．次に関連研究として，単語埋め込みと語彙資源を統合する既存研究を紹介する．具体的には，意味関係および語義の知識を単語埋め込みに統合する事例を紹介する．上位下位関係識別および語義曖昧性解消のタスクに特化した関連研究は，3章および 4章で述べる．

2.1 静的埋め込み

本節では，概念階層への適応による上位下位関係識別（§ 3）で用いる静的埋め込みについて解説する．具体的には，基本となる Word2vec (Mikolov et al. 2013a) および，その拡張形であり本研究で使用する fastText (Bojanowski et al. 2017) について説明する．

2.1.1 概要

静的埋め込みは，語彙 $\mathbb{V}$ に含まれる単語 $w \in \mathbb{V}$ に対して固有の d 次元ベクトル $\boldsymbol{v}_w \in \mathbb{R}^d$ を割り当てることで，単語の意味や性質を表す方法である．ベクトルを決める手段はいくつかあるが，ここで解説する Word2vec は，分布仮説に基づき大規模言語データから人手を介さずに自動的にベクトルを獲得する手法である (岡崎他 2022)．分布仮説とは，単語の意味は周辺に出現する単語，すなわち文脈単語を見ることで類推できるというものである．そこで Word2vec では，言語データから取り出した文を訓練データとして，ある単語から文脈単語を予測する，または文脈単語から中心の単語を予測する問題をニューラルネットによって学習し，得られたパラメータを単語埋め込みとして用いる．使い方が似ている単語は同じような文脈単語が予測されるように学習するので，結果的に文脈類似性が高い単語には似たベクトルが割り当てられる．なおベクトルは単語ごとにひとつ割り当てられるため，多義語の意味は混在するか，暗黙にひとつに絞られる．また語彙に含まれない単語は未知語として無視される．fastText は，部分文字列（サブワード）を語彙に含めることにより未知語を解消した手法である．

Word2vec または fastText を用いると，任意の単語対について cosine 類似度を測定できる．cosine 類似度は，単語間の意味的類似性（例：*money* と *cash*）およびトピック的な関連性（例：*money* と *bank*）と対応することが知られている．また単語ベクトルを加減算することで，ある程度の意味の類推ができる．たとえば $\boldsymbol{v}_{king} - \boldsymbol{v}_{man} + \boldsymbol{v}_{woman}$ は $\boldsymbol{v}_{woman}$ に近くなる．このようにベクトルの加減算があたかも意味の合成に対応するかのような特徴は，加法構成性 (Mikolov et al. 2013a) と呼ばれている．

2.1.2 学習方法

Word2vec のパラメータを学習するモデルは，CBoW: Continuous Bag-of-Words モデルおよび，SG: Skip-Gram モデルの 2 種類がある．両者の違いは，文脈単語から中心の単語を予測する（CBoW）か，中心の単語から文脈単語を予測する（Skip-Gram）かであるが，学習結果には大差ないとされている．ここでは CBoW モデルについて説明する．

言語データから取り出した文の単語列を $w_1 w_2 ... w_t ... w_T$ と表す．位置 t の単語を対象単語 w_t，その前後 ω 個を文脈単語 $C_t = (w_{t-\omega}, ..., w_{t-1}, w_{t+1}, ..., w_{t+\omega})$ と表す．たとえば単語列 *He uses a mouse and keyboard* に対して $t = 4, \omega = 2$ ならば，$w_t = mouse, C_t = \{uses, a, and, keyboard\}$ となる．文脈単語 C_t から対象単語 w_t を予測する問題は，対数尤度

$$\log P(w_t|C_t) = P(w_t|w_{t-\omega}, ..., w_{t-1}, w_{t+1}, ..., w_{t+\omega}) \tag{2.1}$$

を最大化する問題となる．

CBoW モデルでは，$\log P(w_t|C_t)$ を単語埋め込みの内積による対数双線形モデルとして定式化する．具体的には，対象単語 $w_t \in \mathbb{V}$ の単語埋め込みを $\tilde{\boldsymbol{v}}_{w_t} \in \mathbb{R}^d$，文脈単語 $w_{t-\delta} \in C_t; \delta \in \{1, 2, ..., \omega\}$ の単語埋め込みを $\boldsymbol{v}_{w_{t-\delta}} \in \mathbb{R}^d$ と定義して，対象単語と文脈単語の埋め込みの内積をロジットとみなしてソフトマックス関数を適用する．

$$\log P(w_t|C_t) = \boldsymbol{c_t} \cdot \tilde{\boldsymbol{v}}_{w_t} - \log \sum_{w' \in \mathbb{V}} \exp(\boldsymbol{c_t} \cdot \tilde{\boldsymbol{v}}_{w'}) \tag{2.2}$$

$$\boldsymbol{c}_t = \sum_{\delta=1}^{\omega} (\boldsymbol{v}_{w_{t-\delta}} + \boldsymbol{v}_{w_{t+\delta}}) \tag{2.3}$$

なお $\boldsymbol{v}_w$ は $\tilde{\boldsymbol{v}}_w$ と異なることに注意されたい．すなわち単語 w の埋め込みは，対象単語のときは $\tilde{\boldsymbol{v}}_w$ で，文脈単語のときは $\boldsymbol{v}_w$ で与えられる．

式 2.2は，第 2 項がすべての単語 $w' \in \mathbb{V}$ と内積を取る必要があるため，計算負荷が大きいという問題がある．これは $w_t \in \mathbb{V}$ を正解とする $|\mathbb{V}|$ 値分類問題を解いているためである．このため実用上は，NS: Negative Sampling という近似解法が用いられる．NS における対数尤度の計算は，w_t を正例，乱択した k 個の単語集合 $\mathbb{N}_t$ を負例とする二値分類問題によって近似する．具体的には，シグモ

イド関数 $\sigma(.)$ を用いて

$$\log P(w_t|C_t) \approx \log \sigma(\boldsymbol{c_t} \cdot \tilde{\boldsymbol{v}}_{w_t}) - \log \sum_{w' \in \mathbb{N}_t} (1 - \sigma(\boldsymbol{c_t} \cdot \tilde{\boldsymbol{v}}_{w'})) \tag{2.4}$$

と定義する．$\mathbb{N}_t$ は，単語のユニグラム分布 $P(w)$ を平滑化した確率分布から乱択される．

さて $\boldsymbol{v}_w$ および $\tilde{\boldsymbol{v}}_w$ は学習可能パラメータであり，すべての位置 t に対する負の対数尤度の合計

$$L = -\sum_{t=1}^{T} \log P(w_t|C_t) \tag{2.5}$$

を最小化するように最適化する．言語データから逐次的に取り出した文に対して最適化を行い，最終的に得られた $\{\boldsymbol{v}_w\}_{w \in \mathbb{V}}$ を Word2vec の単語埋め込みとする．

Word2vec における語彙は，言語データにおける出現頻度が一定値以上の単語を選定するのが一般的である．このためデータに含まれないか低頻度の単語は未知語として扱われ，ベクトルは割り当てられない．この性質は，新語や専門的な用語を扱う上で問題となりうる．fastText は，語彙に長さ n の部分文字列（サブワード）を追加して，単語をサブワードから組み立て可能にすることで未知語を実質的に解消する手法である．CBoW モデルを用いて具体的に説明すると，文脈単語の埋め込み $\boldsymbol{v}_w$ を，単語 w およびそのサブワード $g \in \mathbb{G}_w$ の埋め込み $\boldsymbol{z}$ を用いて

$$\boldsymbol{v}_w = \boldsymbol{z}_w + \sum_{g \in \mathbb{G}_w} \boldsymbol{z}_g \tag{2.6}$$

と定義する．たとえば $w = where$ で $n = 3$ ならば $\mathbb{G}_w = \{$<wh, whe, her, ere, re>$\}$ となる．なお対象単語の埋め込み $\tilde{\boldsymbol{v}}_w$ ではサブワードは考慮しない．これにより，語彙に含まれない単語 $w' \notin \mathbb{V}$ の埋め込みは $\sum_{g \in \mathbb{G}_{w'}} z_g$ で計算できるようになる．

複合語や前置詞句のような複単語表現の埋め込みは，しばしば構成単語ベクトルの算術平均で与えられる (Mikolov et al. 2013a, Wieting et al. 2015)．算術平均を正当化する根拠は，CBoW モデルが文脈単語に対して合計を取っていること（式 2.2）および，ベクトルの加減算によって意味を合成できる加法構成性の性質（§ 2.1.3）である．複単語表現を単語から与えるという方法論を拡張して，

文の埋め込みを単語埋め込みから構成する方法も提案されている．具体的には，単語埋め込みの算術平均 (Mikolov et al. 2013a, Wieting et al. 2015) や，単語出現頻度にもとづく重み付き和 (Arora et al. 2017, Arroyo-Fernández et al. 2019) といった方法が提案されている．

最後に，訓練データおよび学習時の設定に関する知見を述べる．訓練用の言語データとしては，Wikipedia やニュース記事など，数億から数十億語規模のコーパスが使用されることが多い．コーパスの大規模化は性能向上に寄与するが，一定程度で飽和するとされる (Mikolov et al. 2013b)．語彙数は 100 万，単語ベクトルの次元数は 100–300 が典型的である．文脈単語数 ω は一般に前後 5 単語が用いられる．ω は埋め込みが捉える類似性に影響し，小さくすると意味的類似性，大きくするとトピック的な関連性が強く反映されるといわれる (Levy and Goldberg 2014)．

2.1.3 意味表現としての特徴

埋め込みがどのような特徴を捉えているかを評価する方法は，評判分析や文書分類などの下流タスクの特徴量として用いる方法（Extrinsic Evaluation）および，特定のタスクとは切り離してベクトルと意味の関係性を調べる方法（Intrinsic Evaluation）に大別される (Bakarov 2018)．ここでは Intrinsic Evaluation による研究の紹介を通じて，静的埋め込みが捉える意味の性質を述べる．

ひとつめの評価は，単語埋め込みの cosine 類似度を，人間が付与した意味的類似性およびトピック的な関連性と比較し，cosine 類似度と人手評価の相関を調べるものである．主な評価データセットは，WordSim-353 (Finkelstein et al. 2001) や SimLex-999 (Hill et al. 2015) である．これらのデータセットには，アノテータが判断した類似度や関連度のスコアが付与されている．Word2vec や fastText の cosine 類似度に対するスコアの相関係数は 0.6–0.8 (Wang et al. 2019) であることから，静的埋め込みは，人間が考える意味的類似性およびトピック的な関連性をよく捉えているといえる．一方で人手評価との相関を調べることで示された課題は，対義語（例：*traditional* と *modern*）と，意味的類似性が高い語（例：*traditional* と *conventional*）とを見分けるのが難しいことである (Ono et al. 2015)．これは，対義語が同義語や関連語と似た文脈で用いられるためである (Nguyen et al. 2016)．この問題を緩和するには，語彙知識の利用が有効だと指摘されている．

つまり，人間が決めた近さ・遠さが類似度に反映されるように，埋め込みを変更または更新することである (Faruqui et al. 2015).

ふたつめの評価は，単語埋め込みの加減算を人間が行う類推になぞらえて，単語埋め込みの差が単語間の関係（形態論的または意味的な関係）に対応するかを調べるものである．この評価方法は類推タスク (Mikolov et al. 2013a) と呼ばれている．類推タスクは，同一の関係を持つふたつの単語対 (w_a, w_a^*) および (w_b, w_b^*) から作成される．たとえば $(w_a, w_a^*, w_b, w_b^*) = (\textit{France, Paris, Japan, Tokyo})$ である．この例ではどちらの単語対も，国名と首都という関係にある．単語埋め込みによる類推タスクは $\boldsymbol{v}_{w_b}$ に $\boldsymbol{v}_{w_a} - \boldsymbol{v}_{w_a^*}$ を加算することで w_b に対して (w_a, w_a^*) とおなじ関係にある単語を見つけられるかを調べる．具体的には

$$\hat{w} = \arg\max_{w \in \mathbb{V}} \cos(\boldsymbol{v}_{w_a} - \boldsymbol{v}_{w_a^*} + \boldsymbol{v}_{w_b}, \boldsymbol{v}_w) \tag{2.7}$$

を計算[1] し，$\hat{w} = w_b^*$ ならば正解とする．類推タスクの主な評価データセットは，Google Analogy (Mikolov et al. 2013a) および Microsoft Research Syntactic Analogies (Mikolov et al. 2013c) である．Word2vec や fastText による類推タスクの正解率は 60–80%である (Mikolov et al. 2013a, Bojanowski et al. 2017, Wang et al. 2019) ことから，単語埋め込みの加減算は単語間の関係の合成に対応する，すなわち加法構成性が示唆される．こうした知見は，複単語表現（たとえば *wood mouse*）や文の埋め込みを計算する方法として，しばしば構成単語の算術平均を用いる動機となっている．なお加法構成性の原理は理論的にも分析されており，対象単語と文脈単語の内積（式 2.2）が，両単語の共起確率に対する自己相互情報量を近似するためだと説明されている (Allen and Hospedales 2019, Arora et al. 2016).

類推タスクによる評価は単語埋め込みが単語間の関係を捉えていることを示唆するが，一方で詳細な意味関係の捕捉が課題であることも明らかにしている．具体的には，類推タスクの正解率を関係ごとに精査すると，対義・同義・上位下位関係に対する正解率はほとんど 10%未満だと報告されている (Köper et al. 2015)．この課題を招く要因のひとつは，意味の混在だと指摘されている（Meaning conflation deficiency: Camacho-Collados and Pilehvar (2018)）．すなわち，意味の文脈依存性を捨象して単語ごとにひとつのベクトルを割り当てるという，静的単語埋

[1] この計算式は 3CosAdd (Mikolov et al. 2013c) と呼ばれている．異なる計算式もある．

め込みの原理的な限界である．このような課題の存在は，埋め込みと語彙資源を統合することの意義を主張する根拠のひとつでもある．

2.2 文脈依存埋め込み

本節では，文脈依存埋め込みの計算モデルについて解説する．特に，意味ネットワークへの適応による語義曖昧性解消 (§ 4) で用いる BERT: Bidirectional Encoder Representations from Transformers (Devlin et al. 2019) について詳述する．

2.2.1 概要

文脈依存埋め込みは，単語列 $w_1 w_2 ... w_t ... w_{T-1} w_T$ に含まれる単語 w_t に対して，周辺の単語を考慮したうえで d 次元ベクトル $\boldsymbol{v}_{w_t} \in \mathbb{R}^d$ を計算することで，単語に固有の情報と文脈に含まれる情報の両方を捉える方法である．単語列を埋め込みに変換するモデル（エンコーダともいう）はいくつかあるが，本研究で使用する BERT は，主にマスク言語モデリングを用いて，大規模言語データから自己教師あり学習を行う手法である (岡崎他 2022)．マスク言語モデリングとは，任意の位置の単語を隠して，周辺の単語から隠された単語を予測するタスクであり，直感的には英語試験などに用いられる穴埋め問題に似ている（実際に単語穴埋めタスクとも呼ばれる）．本タスクを用いる動機は，単語予測タスクを解くことは幅広い自然言語処理タスクに対して有効な帰納バイアスを与えるはずだという仮説 (Howard and Ruder 2018) と，その有効性が実験的に示されたことにある (Radford et al. 2018, Howard and Ruder 2018, Peters et al. 2018)．

BERT が採用するモデルアーキテクチャは Transformer (Vaswani et al. 2017) である．Transformer を構成するマルチヘッド注意機構は，周辺の単語との加重平均を取ることで文脈の情報を反映する役割を果たしている．文脈依存埋め込みの計算は，単語列を入力して Transformer の中間層から隠れ状態ベクトルを取り出すことで行われる．隠れ状態ベクトルには単語穴埋めタスクを解くのに必要な情報が含まれるはずである．このため，埋め込みには文脈の情報が反映されるとともに，同じ単語でも文脈に応じて異なるベクトルが計算される．

BERT による文脈依存埋め込みは，語彙，統語，意味論，世界知識に関する情報を含んでいる．埋め込みを取り出す層の特徴については，下位層は単語固有の情報を，上位層は文脈の情報を反映する傾向にある．文脈依存の単語の意

味については，異なる意味や用法ごとに区別が可能であるが，語彙資源における語義の区分ほど詳細ではなく，語義曖昧性解消タスクに用いるには改善の余地があることが示唆されている．

本研究の方法論は，BERT を文脈依存埋め込みのエンコーダとして用いている．これは BERT によって単語列を高次元の特徴量に変換する，特徴抽出のアプローチだといえる．一方で，評判分析や文書分類などの下流タスクの訓練データで BERT のモデルパラメータ自体を微調整するアプローチも存在し，これをファインチューニングと呼ぶ．ファインチューニングは教師あり学習を前提とする転移学習で広く用いられている．

2.2.2 学習方法

BERT は Transformer をアーキテクチャとして採用し，大規模言語データを訓練データとして単語穴埋めおよび次文予測タスクを解くことで，パラメータを学習するモデルである．本来の Transformer は機械翻訳タスクを想定した系列変換モデルであるためにエンコーダ部分とデコーダ部分から構成されているが，BERT は入力文をベクトル列に変換するエンコーダ部分のみを用いている．本節では，まずアーキテクチャの概要を説明してから，次に各タスクの定義を述べる．最後に，訓練に関する知見および，派生モデルについて紹介する．

言語データから取り出した文，すなわち単語列を $w_1 w_2 ... w_t ... w_T$ と表す．より正確に述べると，単語は部分単語に分割される（例：*unrelated* は $(\texttt{un}, \texttt{rel}, \texttt{ated})$ に分割されうる）のだが，以下では単語と部分単語を区別せずに説明する．また単語列にはいくつかの特殊なトークンが付与される．具体的には，先頭には `[CLS]` トークン，文境界（および，一般に文末）には `[SEP]` トークンを挿入する．

Transformer アーキテクチャは，入力埋め込み層 H^I，L 個の中間層 $\{H^l\}_{l=1}^L$，出力埋め込み層 H^O の積み重ねにより構成される．入力埋め込み層は，単語列を d 次元ベクトルの配列 $\boldsymbol{Z}^0 = (\boldsymbol{z}_1^0, \boldsymbol{z}_2^0, ..., \boldsymbol{z}_t^0, ..., \boldsymbol{z}_T^l) \in \mathbb{R}^{d \times T}$ に変換する．

$$\boldsymbol{Z}^0 = H^I(w_1 w_2 ... w_t ... w_T) \tag{2.8}$$

$$\boldsymbol{z}_t^0 = \boldsymbol{e}_{w_t}^{(\mathrm{emb})} + \boldsymbol{e}_t^{(\mathrm{pos})} \tag{2.9}$$

$\boldsymbol{e}_w^{(\mathrm{emb})} \in \mathbb{R}^d$ は単語 $w \in \mathbb{V}$ の埋め込み，$\boldsymbol{e}_t^{(\mathrm{pos})} \in \mathbb{R}^d$ は位置符号と呼ばれるパラメータである．位置符号は学習可能なパラメータではなく，正弦波によって計算

される (Vaswani et al. 2017).

中間層は，$l-1$ 層目の出力 $\boldsymbol{Z}^{l-1}$ を l 層目に入力して $\boldsymbol{Z}^{l}$ に変換することを繰り返す.

$$\boldsymbol{Z}^{l} = H^{l}(\boldsymbol{Z}^{l-1}) \tag{2.10}$$

中間層は，マルチヘッド注意機構，フィードフォワード層，残差結合，層正規化から構成されており，前者 2 つが学習可能なパラメータを含んでいる．またマルチヘッド注意機構は，周辺の単語との加重平均を取ることで，それぞれの単語に文脈の情報を与える役割を果たしている.

注意機構が周辺の単語と加重平均を取る仕組みは，単語 $\boldsymbol{z}_t$ ごとにクエリベクトル $\boldsymbol{q}_t$，キーベクトル $\boldsymbol{k}_t$，およびバリューベクトル $\boldsymbol{x}_t$ を計算し，単語 t からみた単語 u に対する重み $a_{t,u}$ を，平滑化係数 c を掛けたキーとバリューの内積 $s_{t,u} = c\boldsymbol{q}_t^{\mathrm{T}}\boldsymbol{k}_u$ によって決めることで実現されている．具体的には，層番号 l を省略して表記すると，

$$\boldsymbol{q}_t = \boldsymbol{W}^{(Q)}\boldsymbol{z}_t,\ \boldsymbol{k}_t = \boldsymbol{W}^{(K)}\boldsymbol{z}_t,\ \boldsymbol{x}_t = \boldsymbol{W}^{(X)}\boldsymbol{z}_t \tag{2.11}$$

$$s_{t,u} = c\boldsymbol{q}_t^{\mathrm{T}}\boldsymbol{k}_u \tag{2.12}$$

$$a_{t,u} = \frac{\exp(s_{t,u})}{\sum_{t'=1}^{T}\exp(s_{t,t'})} \tag{2.13}$$

$$\hat{\boldsymbol{z}}_t = \sum_{u=1}^{T} a_{t,u}\boldsymbol{x}_u \tag{2.14}$$

と定義される．ここで $\hat{\boldsymbol{z}}_t$ は，周囲の単語と加重平均を取ったあとの表現である．重み a の元をたどると中間層の出力 $\boldsymbol{Z}$ に行き着くことから，注意機構には周辺の単語からどの程度の情報を取り込むかを学習する仕組みが備わっているといえる．またマルチヘッド注意機構は，このような注意機構を複数個並列に並べた機構である.

出力埋め込み層は，各単語の隠れ状態ベクトル $\boldsymbol{z}_t^L$ をロジット $\boldsymbol{s}_t \in \mathbb{R}^{|\mathbb{V}|}$ に変

換する．

$$\boldsymbol{S} = H^O(\boldsymbol{Z}^L) \tag{2.15}$$

$$\boldsymbol{s}_t = \boldsymbol{E}^{(\text{logit})^{\mathrm{T}}} \boldsymbol{z}_t^L \tag{2.16}$$

$\boldsymbol{E}^{(\text{logit})} = \{\boldsymbol{e}_w^{\text{logit}}\}_{w \in \mathbb{V}}; \boldsymbol{e}_w^{\text{logit}} \in \mathbb{R}^d$ は，隠れ状態と内積を取ることで単語 $w \in \mathbb{V}$ に対するロジットを計算するベクトルである．またロジットにソフトマックス関数を適用すると，位置 t における単語の確率分布 $p(w_t|w_1w_2...w_t...w_T)$ が得られる．

$$p(w_t|w_1w_2...w_t...w_T) = \frac{\exp(s_{t,w})}{\sum_{w' \in \mathbb{V}} \exp(s_{t,w'})} \tag{2.17}$$

BERT による文脈依存埋め込み $\boldsymbol{v}_{w_t}$ は，中間層の隠れ状態ベクトルを用いる．どの層を用いるのがよいかはタスク依存であるが，経験的には上位の層は文脈の意味を反映する度合いが強いとされている．具体例として，最上位 4 層の合計を取る場合の式を以下に示す．

$$\boldsymbol{v}_{w_t} = \sum_{l=L-3}^{L} \boldsymbol{z}_t^l \tag{2.18}$$

なお単語が部分単語に分割されている場合は，部分単語の平均を $\boldsymbol{z}_t^l$ として用いることが一般的である．また文の埋め込みを計算する場合は，単語埋め込みを単純に平均する方法 (Toshniwal et al. 2020) や，特殊トークン `[CLS]` をそのまま使用する，またはファインチューニングする方法 (Reimers and Gurevych 2019) がある．

次に，BERT の自己教師あり学習に用いるタスクについて説明する．ひとつめのタスクは単語穴埋めである．このタスクでは，文中でマスクされた単語を予測する．具体的には，乱択で選んだ位置 t の単語 w_t を特殊トークン `[MASK]` に置換した単語列 $w_1w_2...w_t...w_T$ を入力し，該当位置に対して正解単語 w_t が出現する確率の対数尤度を最大化する．

$$\log p(w_t|w_1w_2...\texttt{[MASK]}...w_T) \tag{2.19}$$

マスクされる単語の割合は 15%である．なお `[MASK]` は，単語埋め込みを計算す

る用途や下流タスクを解く用途のときは出現しない．そこで自己教師あり学習のために単語をマスクするときには，80%の確率で [MASK] に置換，10%の確率でランダムな単語に置換，10%の確率で何もしないという操作が行われる．

ふたつめのタスクは次文予測である．このタスクでは，連結して入力された2文の抽出元が同じ文書か否かを識別するタスクである．このタスクは，単語列の先頭 $t = 0$ に付与した特殊トークン [CLS] の埋め込み $\boldsymbol{z}_{\boldsymbol{0}}^{L}$ を入力とする二値分類器を学習することで訓練される．ただし後続の研究では，このタスクはトピックの違いに注目しやすいため必ずしも有用でないと報告されている．実際に，性能改善を目的とした派生モデル (Liu et al. 2019) では省略されることがある．

訓練データおよび学習時の設定に関する知見を述べる．訓練用の言語データとしては，Wikipedia，ニュース記事，書籍，ウェブページなど，数十億から数百億語規模のコーパスが使用されることが多い．語彙数は1万から10万，隠れ状態ベクトルの次元数は数百から千程度，中間層の数は10から20程度が用いられる．また言語データの規模，中間層の数，および語彙数を大規模化することは，いずれも性能向上に寄与することが知られている (Liu et al. 2019)．

BERT は幅広いタスクに対して顕著な効果を発揮したため，多くの派生モデルが研究されている．派生研究におけるひとつの方向性は，Wikipedia などを情報源として世界知識を強化することである．具体的には，人物や地域などのエンティティに対する知識（例：“ボブ・ディランは歌手である”）を問うタスクを単語穴埋めタスクに統合することで，固有表現認識タスクなどの性能が改善することが報告されている (Zhang et al. 2019, Peters et al. 2019)．これらの研究は本研究のテーマである語彙知識との統合とは異なるものの，人間が体系化した記号的知識との統合という観点で関連性がある．もうひとつの方向性は，多言語化である．自己教師あり学習の訓練データとして，多言語テキストコーパスを用いる方法 (Conneau et al. 2020) や，対訳コーパスを用いる方法 (Feng et al. 2022) が提案されている．多言語化モデルは，異なる言語を同一の空間に埋め込めるだけでなく，言語をまたいだ意味的類似性，すなわち語義や文の内容が似ている場合は類似の埋め込みが得られることが報告されている (Cao et al. 2020)．これらの研究は，本研究を発展させて多言語化に取り組む場合の基盤となりうる．

2.2.3 意味表現としての特徴

BERT の埋め込みがどのような特徴を捉えているかを調査する研究は BERTology (Rogers et al. 2020) と総称されており，その調査対象は語彙・統語・意味論・世界知識など多岐にわたる．調査に用いられる主な方法は，静的単語埋め込みと同様に，Intrinsic Evaluation および Extrinsic Evaluation である．また Transformer は複数の層からなるため，隠れ状態ベクトルを取り出す層の違いによる影響もしばしば分析される．本節では，層の違いによる影響および統語・意味論的な情報について概説したのちに，単語の意味を捉える性質について述べる．

層の違いによる影響については，下位の層は単語固有の情報，上位の層は文脈の情報を反映する傾向にある (Vulić et al. 2020, Voita et al. 2019)．最適な層はタスク依存ではあるが，固有表現認識や語義曖昧性解消など，文脈依存の単語の意味を問うタスクでは，最上位の 4 層を使うことが多い (Wang et al. 2020, Bevilacqua and Navigli 2020, Wang and Wang 2020, Loureiro and Jorge 2019, Devlin et al. 2019)．特に，中間層は統語的な情報が顕著であることが広く認められている (Rogers et al. 2020)．具体的には，主に Extrinsic Evaluation を用いて，品詞，句，構文木の情報が復元できることが報告されている (Tenney et al. 2019, Liu et al. 2019)．意味論的な知識については，エンティティの種類およびエンティティ間の関係や，意味役割の情報が捉えられているとされる (Tenney et al. 2019)．

文脈依存の単語の意味を捉える性質については，Insrinsic Evaluation によって，詳細ではないものの疎い粒度で語義を区別できることが報告されている．具体的には，さまざまな用例における多義語の埋め込みを低次元に射影して可視化すると，必ずしも語彙資源で定義される語義の区分とは一致しないものの，異なる意味や用法ごとにクラスタを形成することが報告されている (Reif et al. 2019)．一方で，文脈依存の意味を捉えるのが困難な事例についても報告がある．具体的には，文脈内に複数のトピックを示唆するキーワードが存在する場合や，文節をまたいで周辺単語の情報を取り込んでしまう場合である (Loureiro et al. 2021)．こうした事例を裏付けるものとして，同一文内の単語ペアは乱択した単語ペアよりも類似度が高い，つまり単語固有の情報よりも文脈の情報を反映しすぎることを示唆する実験結果もある (Ethayarajh 2019)．さらに，教師あり語義曖昧性解消タスクでは，埋め込みを用いて対象語に最も似ている用例を検索してそ

の注釈語義を回答する手法が存在するが，おなじ教師ありアプローチでも，識別器を学習する手法には大きく劣る (Devlin et al. 2019, Wiedemann et al. 2019)．これらの報告は，埋め込みが語義を捉える性質には改善の余地があり，したがって語彙資源を統合する意義があることを示唆している．

2.3 語彙資源

本節では，提案手法が使用する語彙資源である WordNet について，語彙知識の種類および規模を解説する．具体的には，WordNet の構造および統計量を説明する．また，WordNet と関連性が深い言語資源についても簡単に紹介する．これらの資源は上位下位関係識別および語義曖昧性解消タスクでもしばしば用いられるため，本研究とも関連する．

2.3.1 WordNet

概要

Princeton WordNet (Jurafsky and Martin 2009, Fellbaum 1998)[2] または単に WordNet は，英語の単語について，単語が取り得る語義および語義どうしの意味関係を体系的に整理した辞書様式のデータベースである．WordNet は 16 万語の名詞・動詞・形容詞・副詞について，12 万の語彙概念および 21 万の語義を整理しており，その網羅性およびリッチな情報によって，概念・意味に関する知識源としてさまざまな研究で使用されている．

構成および意味関係

WordNet を構成する主な要素は，Lemma，Sense，Synset，Gloss，Example である．

- Lemma（レンマ）は正規化かつ品詞で区別された単語であり，いわゆる見出し語である．Lemma は必ずしも一単語ではなく，複合語 *wood mouse* や動詞句 *look up* のような複単語表現も含まれる．
- Sense は Lemma を Sense key で区別したものであり，語義のことである．たとえば名詞 *computer* における “計算機” の語義は `computer%1:06:00::` である．Sense はひとつの Synset に属する．複数の Sense を持つ Lemma は多義語で，単一の Sense を持つ Lemma は単義語である．また同じ Lemma に紐付く Sense は互いに異義（同じ単語の異なる語義）である．

[2]本節の内容は Princeton WordNet version 3.0 にもとづく．本バージョンは語義曖昧性解消タスクで用いる標準的な意味目録として普及している (Raganato et al. 2017).

- Synset (synonym set) は単語と対応づけられた概念，すなわち語彙概念（lexical concept）であり，具体的には同義とみなせる Sense の集合である．おなじ Synset に属する Sense は互いに同義関係であり，その Sense に紐付く Lemma は互いに同義語である．

- Gloss は Synset を自然言語で説明するテキストであり，辞書でいう語釈文や定義文である．

- Example は Synset の使われ方を自然言語で表した数単語の短文であり，いわゆる例文である．Example は必ず利用できるわけではなく，Synset のうち 28%にのみ付与されている．また短文であるため文脈情報は乏しい．

WordNet では Lemma がいわゆる索引の役割を果たしており，Lemma で検索するとその単語が取りうる語義が Sense の配列として返ってくる．なお Sense はおおむね出現頻度順に整列されている (Jurafsky and Martin 2009)．このため配列先頭の Sense は WordNet first sense とも呼ばれ，いわゆる多数決法によって語義曖昧性解消を行う場合の予測語義として使用されている．

表 2.1に，語義 `mouse%1:06:00::` に対する要素を例示する．この語義は，名詞 *mouse* における “入力機器のマウス” の意味に対応する．

表 2.1: 語義 `mouse%1:06:00::` に対する WordNet の要素

要素	値
Sense	`mouse%1:06:00::`
Lemma	*mouse*
Synset	`mouse.n.04`
Gloss	*a hand-operated electronic device ...*
Example	*a mouse takes much more room than a trackball*

表 2.1で示した要素のほかに，Lexicographer category または Supersense（超語義）と呼ばれる要素がある．これは，45 種類の高度に抽象化された概念の一覧であり，ほぼすべての Synset はひとつの Supersense に属する．たとえば `mouse.n.04` の Supersense は `noun.artifact` である．もともとは辞書編纂作業の分業化を目的として，概念を詳細化してゆく起点として定義されたものであるが，語義曖昧性解消の研究などでは粗粒度の意味目録としてもしばしば用いられる．

Synset および Sense は，Semantic relation（意味関係）で接続されている．表 2.2に主な意味関係の一覧を示す．具体例の欄は，X および Y を Synset または Sense として，X に対して Y が特定の意味関係にあることを X→Y と表記している．ただしわかりやすさのため，Synset ID や Sense key ではなく単語で例示した．なお日本語による定義は，林 (2010) および日本語 WordNet ウェブサイト[3] を参考にした．

表 2.2: 意味関係の一覧

意味関係	品詞	定義	具体例（X→Y）
Hypernym	名詞，動詞	X は Y に包含される	*mouse→rodent*
Hyponym	名詞，動詞	X は Y を包含する	*rodent→mouse*
Holonym—Part	名詞	X は Y を構成する	*China→Asia*
Meronym—Part	名詞	X は Y によって構成される	*Asia→China*
Holonym—Member	名詞	X は Y の構成員である	*Canis→Canidae*
Meronym—Member	名詞	X は Y を構成員とする	*Canidae→Canis*
Holonym—Substance	名詞	物質/材料 X は Y を構成する	*oxygen→ozone*
Meronym—Substance	名詞	X は物質/材料 Y で構成される	*ozone→oxygen*
Domain—Topic	すべて	X はトピック Y に属する	*comet→astronomy*
In Domain—Topic	名詞	X が Y の属するトピックである	*astronomy→comet*
Domain—Usage	すべて	X の用法は Y に限定される	*jean→plural form*
In Domain—Usage	名詞	X は Y の用法を規定する	*plural form→jean*
Domain—Region	すべて	X は地域 Y に属する	*karate→Japan*
In Domain—Region	名詞	X が Y の属する地域である	*Japan→karate*
Instances	名詞	X は Y の具体例である	*B-52→bomber*
Has Instance	名詞	X は Y を具体例にもつ	*bomber→B-52*
Entails	動詞	X を行うとき，必ず Y も行う	*snore→sleep*
Causes	動詞	X を行うと，Y を引き起こす	*kill→die*
See also	動詞，形容詞	X と Y の間に何らかの関連がある	*accurate→precise*
Attribute	名詞，形容詞	X が属性 Y を表す際に使われる	*large→size*
Similar to	形容詞	X の表す意味が Y と近似している	*large→bulky*
Synonym (Sense)	すべて	X と Y は同じ意味である	*water→H2O*
Antonym (Sense)	すべて	X は Y の反対の意味を持つ	*able→unable*
Pertainym (Sense)	形容詞，副詞	X は Y の形容詞・副詞に対応する	*chaotic→chaos*

意味関係のうち，Synonym（同義），Antonym（対義），Pertainym（分詞形容詞）は単語レベルで定義されるため，Sense どうしの関係である．その他は Synset どうしの関係である．また，語形変化という形態論的な関係に対しては，Sense どうしで Derivationally Related Form という関係が定義されている．たとえば *compute* は *computer* の動詞化であり，両者は形態論的な関係がある．これは意味関係ではないため表 2.2には含めていないが，形態論的な関係は，品詞を

[3]https://bond-lab.github.io/wnja/ 2023 年 8 月 19 日アクセス

またいだ意味的関連性を把握するうえで有用である．

意味の階層構造

階層構造をなす，すなわち推移律および反対称律を満たす意味関係は，Hypernym/Hyponym による上位下位関係（hyponymy）および，Holonym/Meromym による全体部分関係（meronymy）のふたつが代表的である．これらの意味関係は名詞および動詞に特有であり，特に名詞的概念の体系化において中心的な役割を果たしている．形容詞および副詞には存在しない．また上位下位関係により形成される階層構造のことを，特にタクソノミ（taxonomy）と呼ぶ (林 2010)．

上位下位関係と全体部分関係の大きな違いは，前者は抽象化と具象化，後者は合成と分解という操作と対応することである．上位下位関係の重要な特徴は，親概念の性質を継承することである．たとえば *robin* は Hypernym である *bird* が持っている「空を飛ぶ」性質を継承する．これに対して全体部分関係は，一般に親概念の性質は継承しない．たとえば *pedal* は Holonym である *bicycle* が持っている「乗り物」という性質は持っていない．親概念の性質に制約されないという特徴は，全体部分関係がしばしば多数の親概念をとることにも現れる．たとえば *handle* は *umbrella,brush,teacup,...* を Holonym にとる．こうした理由から，テキスト間の論理関係推論，文書要約，情報検索といった応用を見据える場合は，全体部分関係よりも上位下位関係のほうが興味深い．また上位下位関係識別タスクにおいては，文字通り，Hypernym/Hyponym による関係か否かを判定することが要求される．なお全体部分関係を細分化すると，Member（集合），Part（構造），Substance（物質/材料）による関係に分けられる．Member は生物学的な分類体系，Part は地理・数量単位・生物学的な構造，Substance は物質/材料を主に記述している．

動詞における上位下位関係について，Entails（含意）および Causes（因果）と対比しながら説明する．動詞の上位下位関係の定義は troponymy（様態）という考え方に基づいており，自然言語では *To X is to Y in some particular manner* と表現される (Huminski and Zhang 2018)．たとえば *to whisper* は *to talk* を声量という様態で特化した関係にあるため，troponym である．これに対して Entails は，X を行うときに必ず同時に Y も行っているという関係である．たとえば *snore* する（いびきをかく）ときには必ず *sleep* する（眠る）ため，Entails である．こ

れらの例からもわかるとおり，Troponym ならば Entails でもあるが，逆は成り立たない．つまり動詞では，含意関係の特別な場合が上位下位関係である．Causes は原因と結果の関係である．たとえば *kill* は *die* を引き起こす原因である．なお Causes は（すくなくとも WordNet では）階層構造ではない．Entails は 4 層程度の浅い階層構造である．

上位下位関係は階層構造を形成して推移律および反対称律を満たすが，データ構造としては必ずしも木構造ではない．たとえば複数の Hypernym を持つ Synset も存在するし，動詞の Hyponym は単一の起点から分岐していない，すなわち全概念を包含する唯一の Synset は定義されていない．また当然のことながら，単語レベルでは Sense ごとに異なる単語と上位下位語ペアを形成する．

意味のネットワーク構造

WordNet における意味のネットワークは，Lemma，Sense，Synset およびそれらを結ぶ意味関係によって構成される．ただしここでは，おなじ Synset に属する Sense どうしを同義関係で結び，Lemma は正規化かつ品詞で区別した単語として解釈しよう．すると意味のネットワークは，単語および Sense（語義）をノード，単語の意味と語義間の意味関係をエッジとするネットワークとみなせる．

ほとんどの意味関係（表 4.2）は，意味的またはトピック的な関連性がある語義を結んでいる．したがって，意味ネットワークで互いに隣接する語義（例：“酸素”—“オゾン”）は，意味的な類似性が高い．またこれらは文内で共起しやすい，あるいはしばしば置き換え可能な単語の語義でもあり，その点では分布仮説における文脈の類似性と共通する部分がある．ただし Antonym は対義関係，Domain—Usage は形態論的制約を表すように，意味的類似性があてはまらないものもある．逆に，ネットワークで隣接せず 2 ホップ以上離れている語義は，上位下位関係のみを辿るような特別な場合を除き，意味的な関連性や類似性が低いといえる．

単語に隣接する語義は，単語が取りうる意味を表している．したがって単義語はひとつの語義と，多義語は複数の語義と隣接する．また単語の近傍，すなわち単語に隣接する語義の集合は，同じ単語の互いに異なる意味の集まりである．

エッジすなわち意味関係の向きについては，大半が双方向である．具体的には表 2.2の具体例からもわかるとおり，Hypernym/Hyponym，Meronym/Holonym，

表 2.3: WordNet を構成する要素の統計量

要素	名詞	動詞	形容詞	副詞	合計
Synset 数	82,115	13,767	18,156	3,621	117,659
Gloss 数	82,115	13,767	18,156	3,621	117,659
Lemma 数	117,798	11,529	21,479	4,481	155,287
うち，多義語	15,938	5,252	4,976	733	26,899
Sense 数	146,320	25,047	30,002	5,580	206,949
Synset 平均	1.8	1.8	1.7	1.5	1.8
Lemma 平均	1.2	2.2	1.4	1.2	1.3
Lemma(多義語) 平均	9.2	4.8	6.0	7.6	7.7
Example 数	11,489	12,528	20,182	4,140	48,339
Synset 平均	0.1	0.9	1.1	1.1	0.4
意味関係の数	231,153	29,480	34,927	4,042	299,602
Synset 平均	2.8	2.1	1.9	1.1	2.5

Domain/In Domain，Instance/Has Instance は互いの向きを反転させた関係にある．たとえば *mouse→rodent* が Hypernym ならば，*rodent→mouse* は Hyponym である．また See also，Similar to，Antonym はもともと双方向の関係である．品詞間のつながりについては，ほとんどの意味関係はおなじ品詞どうしを結んでおり，異なる品詞どうしを結ぶ意味関係は，Domain が名詞とそれ以外，Attribute が形容詞と名詞，Pertainym が形容詞と副詞を結ぶ程度である．したがって，意味ネットワークは品詞内で密なサブネットワークを形成しており，サブネットワーク間のつながりは限定的だといえる．なおそれぞれの品詞における中心的な意味関係は，名詞および動詞は Hypernym/Hyponym，形容詞は Similar to，副詞は Pertainym である．意味関係の数については後述の表 2.4を参照されたい．

統計量

WordNet の規模を説明するため，Lemma，Sense，Synset，意味関係の数および各種要素ごとの平均値を，表 2.3に示す．なお Gloss は Synset に対して必ずひとつ付与されているため，両者の数は一致する．

WordNet に採録されている Lemma（単語），Sense（語義），および Synset（語彙概念）は，それぞれ 16 万語，21 万件，および 12 万件である．Synset ごとの語義数，すなわち同義語数は平均 1.8 語である．単語別の語義数は平均 1.3 件だが，そもそも複数の Sense を持つ多義語が 2 割弱にとどまるため，多義語に限

定すると語義数は平均 7.7 件に急増する．したがって WordNet を意味目録（語義の一覧）とする語義曖昧性解消タスクは，平均して約 8 件の候補から正解を選ぶ識別タスクだといえる．Example（例文）の数は Synset の半分未満であり，特に名詞においては平均 0.1 件と非常に乏しい．なお Gloss および Example を合計したテキストの規模は約 160 万語である (Rademaker et al. 2019)．品詞別では名詞が 7 割を占めており，意味関係の数や多義語の数も名詞がもっとも多いことから，名詞を中核とした構成であることがわかる．動詞は語義数が平均 2.2 かつ意味関係の数が平均 2.1 と比較的に多く，密なネットワークを形成しているといえる．

語義どうしの関連性の規模を例示するため，Synset と Sense が持つ意味関係の数を，表 2.4に示す．また，参考情報として形態論的関係である Derivationally Related Form の数も同表に示す．

WordNet に含まれる主な意味関係の数は，Hypernym/Hyponym が 8.9 万件，Meronyms/Holonyms が 2.2 万件，Similar to が 2.1 万件，Pertainym が 0.9 万件である．したがって各品詞において語義を体系化する中心的な役割を果たしている意味関係は，名詞および動詞は主に上位下位関係による概念階層，形容詞は Similar to による類似性，副詞は Pertainym による名詞・形容詞との対応であることが確認できる．なお § 2.3.1にて意味関係は品詞内で密なサブネットワークを形成していると述べたが，品詞をまたいだ結びつきは，Domain や Attribute を除くと，形態論的関係である Derivationally Related Form が支配的であることがわかる．このような特徴から，意味関係的知識を応用する場合の示唆がふたつ得られる．具体的には，タスクによっては概念階層をなす上位下位関係だけでなく多様な意味関係を活用すべきであること，形態論的関係などを経由して名詞・動詞の豊富な情報を他の品詞に波及させることである．

2.3.2 WordNet に関連する言語資源

本節では，WordNet と関連性が深い言語資源として，上位下位関係識別および語義曖昧性解消のベンチマークとして使用されるデータセットおよび，語義注釈付き用例文コーパスである SemCor，多言語かつ固有表現に拡張された語彙資源である BabelNet について紹介する．

表 2.4: 意味関係を持つ Synset または Sense の数

意味関係	名詞	動詞	形容詞	副詞	合計
Hypernym	75,850	13,239	0	0	89,089
Hyponym	75,850	13,239	0	0	89,089
Holonym—Part	9,097	0	0	0	9,097
Meronym—Part	9,097	0	0	0	9,097
Holonym—Member	12,293	0	0	0	12,293
Meronym—Member	12,293	0	0	0	12,293
Holonym—Substance	797	0	0	0	797
Meronym—Substance	797	0	0	0	797
Domain—Topic	4,250	1,257	1,099	37	6,643
In Domain—Topic	6,643	0	0	0	6,643
Domain—Usage	660	15	220	72	967
In Domain—Usage	967	0	0	0	967
Domain—Region	1,269	2	73	1	1,345
In Domain—Region	1,345	0	0	0	1,345
Instances	8,577	0	0	0	8,577
Has Instance	8,577	0	0	0	8,577
Entails	0	408	0	0	408
Causes	0	220	0	0	220
See also	0	7	2,685	0	2,692
Attribute	639	0	639	0	1,278
Similar to	0	0	21,386	0	21,386
Antonym (Sense)	2,152	1,093	4,024	710	7,979
Pertainym (Sense)	0	0	4,801	3,222	8,023
合計	231,153	29,480	34,927	4,042	299,602
Derivationally Rel. Form	37,250	23,134	14,332	1	74,717

BLESS

BLESS (Baroni and Lenci 2011) は，意味関係識別タスクのベンチマーク用データセットであり，上位下位関係識別タスクの評価データセットとして広く使用されている (Nguyen et al. 2017, Vulic and Mrksic 2018, Nguyen et al. 2017). BLESS には，生物・無生物合計 200 個の名詞に対して，上位・全体・同位・属性などの関係にある名詞・動詞・形容詞が単語対として収録されている．文脈は付与されていないが，そのかわりに多義性が顕著でない単語が採録されている．意味関係のアノテーションは，主に WordNet に基づいている．データセットのサイズは 26,554 件で，そのうち上位下位語対は 1,343 件である．

HyperLex

HyperLex (Vulic et al. 2017) は，単語間含意関係（Lexical entailment）タスクのベンチマーク用データセットであり，上位下位関係識別研究における評価データセットとして使用されている．HyperLex が想定するタスクは識別ではなくランキング，すなわち与えられた単語ペアを上位下位関係らしさの高い順に順位付けして，アノテータが付与した順位との一致度を評価するタスク (Graded lexical entailment) である．このため，HyperLex における上位下位関係のアノテーションは，二値ではなく 0–10 のスコア，具体的には *To what degree is X a type of Y?* という問いで定量化した関係として扱われている．収録されている単語対は，品詞は名詞または動詞，意味関係は上位下位・全体部分・同位・同義・対義・無関連である．単語対は主に WordNet から採録されている．データセットのサイズは 2,616 件である．

Unified Evaluation Framework for WSD

Unified Evaluation Framework for Word Sense Disambiguation (Raganato et al. 2017) は，語義曖昧性解消タスクのベンチマーク用データセットである．1998 年に開始された Senseval ワークショップおよび後継である SemEval ワークショップでは，語義曖昧性解消タスクの性能評価が継続的に行われてきた．本データセットはこれらのワークショップで構築された評価データセットを，WordNet を意味目録として整理・統合したものである．元になった評価データセットは，Senseeval-2 (Edmonds and Cotton 2001)，Senseeval-3 task 1 (Snyder and Palmer 2004)，SemEval-07 task 17 (Pradhan et al. 2007)，SemEval-13 task 12 (Navigli et al. 2013)，SemEval-15 task 13 (Moro and Navigli 2015) の 5 種類である．本データセットでは，用例文の各単語（単義語を含む）に対して，WordNet の Lemma および Sense がアノテーションされている．したがって，Lemma を検索して得られた Sense の配列からひとつを選択し，それを正解と比較することで語義曖昧性解消の精度を定量化できる．対象の品詞は名詞・動詞・形容詞・副詞の 4 種類で，アノテーション数は 7,253 件である．

WSD Hard Benchmark

WSD Hard Benchmark (Maru et al. 2022) は，語義曖昧性解消タスクの新たなベンチマーク用データセットである．事実上の標準である Unified Evaluation Framework for WSD における教師あり WSD の精度は，2020 年に人間と同等と目される 80%に到達した (Bevilacqua and Navigli 2020)．しかし個別事例に着目すると人間では考えがたい自明な誤りが依然として存在しており，精度を額面通りに解釈する妥当性をめぐって議論がされている．また既存の言語資源は，評価データセットを含めて，使用頻度が高い一般的な語義に偏っているという指摘がある (Pasini 2020)．かかる背景のもと，本データセットは，非一般的な語義に対する評価，評価の正確性の改善，およびタスクの高難度化を目的として構築された．具体的には，本データセットは 5 つのサブセットから構成されている．まず，42D である．このサブセットは使用頻度が低い語義を正解とする，具体的には 1) SemCor に出現しない かつ 2) WordNet first sense ではない 語義を正解とする事例である．したがって 42D による評価は，非一般的な語義における性能を調べるうえで有用である．次に，$\mathrm{ALL}_{\mathrm{NEW}}$ および $\mathrm{S10}_{\mathrm{NEW}}$ である．これらは既存のデータセットである Unified Evaluation Framework for WSD および SemEval-2010 task 17 (Agirre et al. 2010) のアノテーション誤りを修正したものである．これらは評価の正確性の改善に寄与するものである．最後に，$\mathrm{ALL}_{\mathrm{NEW}}$・$\mathrm{S10}_{\mathrm{NEW}}$・42D に採録された事例を再編して作成した hardEN および softEN である．具体的には，hardEN は，データセット構築時点における主な教師あり WSD 手法のいずれでも解けない事例のみを集めたものである．softEN は hardEN の残りである．したがって hardEN による評価は，高難度の事例における性能，および既存手法に対する優位性を調べるうえで有用である．

データセット構築に加えて，Maru et al. (2022) は従来のマイクロ F 値よりもマクロ F 値を適切な評価指標として推奨した．また 42D サブセットを用いて主な既存手法の性能を評価し，最高精度は 59%と，従来のベンチマークにおける 80%超よりも 20 ポイントも低いことを示した．これにより，既存手法は総じて一般的な語義を選好し，非一般的な語義の識別を苦手とする傾向があることを明らかにした．

SemCor

SemCor (Semantic Concordance) (Miller et al. 1993) は，WordNet における用例情報の不足を補うために開発された語義注釈付き用例文コーパスであり，本論文執筆時点では教師あり WSD アプローチにおける標準的な訓練データセットとして使用されている (Raganato et al. 2017)．SemCor は，均衡コーパスである Brown Corpus から抽出した文の各単語に対して，WordNet の Lemma および Sense を付与している．すなわち，WordNet を意味目録として，単語に対する語義のアノテーションが行われている．SemCor が提供する文脈情報は，人手で作成された語義注釈付き用例文コーパスの中では最大規模である．具体的には，コーパスの単語数は約 80 万語，アノテーション数は 22.6 万件，アノテーションされた語義の異なり数は 3.3 万件である．WordNet にも例文（Example）は存在するが，数単語の短文が 4.8 万件のみであることから，SemCor のほうが明らかに大きい．一方で WordNet の語義数は 21 万件であることを考えると，最大規模といえども語義カバレッジは約 15%にすぎないこともわかる．SemCor は均衡コーパスからテキストを収集しているため，日常的な言語使用が反映された結果，使用する機会が少ない単語や語義のカバレッジが低いのだと指摘されている (Maru et al. 2022).

Coarse Sense Inventory

Coarse Sense Inventory (CSI) (Lacerra et al. 2020) は，WordNet の Synset（概念）を 45 個の高度に抽象化された意味カテゴリに整理し直した意味目録である．CSI は 8.3 万件の Synset（全体の約 7 割）について，ひとつ以上の意味カテゴリを付与している．たとえば“計算機”の概念は `CRAFT_ENGINEERING_AND_TECHNOLOGY_` という意味カテゴリに属する．CSI の目的は，WordNet における語義の区別がしばしば人間にも識別不能なほど細かすぎるという批判に対処しつつ，Supersense（§ 2.3.1）よりも実用的に優れた意味目録を提供することである．CSI は語義曖昧性解消タスクにおける意味目録，またはタスクを解く際の語彙知識として使用されている (Wang and Wang 2021).

BabelNet

BabelNet (Navigli et al. 2021) は，WordNet を基盤とした多言語語彙資源であり，機械翻訳，知識グラフ，語義曖昧性解消などを多言語化する研究で広く使用されている．BabelNet は，各言語の WordNet を統合することにより，特定の概念をさまざまな言語の語彙項目と紐付けている．たとえば “動物のイヌ” に対応する Synset は，*chien*（仏），*cane*（伊），*hund*（独）などの Lemma と紐付いている．こうした情報は，多言語間での正確な意味の対応付けをする場合に有用である．また BabelNet は，Wikipedia および Wikipdata を統合することにより，固有表現の網羅性を大きく改善している．たとえば “戦闘機” の Synset に対して Instances の関係にある固有表現は，WordNet では *B-52* の 1 件のみだが，BabelNet では 30 件以上が収録されている．こうした情報は，テキストを知識ベースと対応づけて解析する場合に有用であるほか，語義曖昧性解消とエンティティリンキングを統合する場合にも活用できる．

分類語彙表

分類語彙表 (国立国語研究所 2004, 柏野 2006) は，日本語の単語を意味ごとに分類・整理したシソーラスである．分類語彙表では階層的な意味的範疇（意味の範囲）が採用されており，具体的には類・部門・中項目・分類項目の 4 階層から成り立っている．また最下層の分類項目には，同義または類義の関係にある見出し語が含まれている．さらに分類項目には 5 桁の分類番号が割り当てられ，番号の先頭桁が上位の意味的範疇を表す構造になっている．たとえば分類項目<哺乳類>は，<体>←<自然>←<動物>←<哺乳類> の階層に位置しており，見出し語は**獣，動物，人類，コアラ，猫**などが含まれている．分類番号は 1.5501 であり，先頭 3 桁の 1，1.5，1.55 がそれぞれ<体>，<自然>，<動物>を表している．なお意味的範疇を山括弧<>，下位範疇から上位範疇への関係を左矢印 ←，見出し語を**ゴシック体**で示した．分類項目数は約 900，見出し語の異なり数は約 8 万語である．後者は前者よりも 2 桁大きいこと，および見出し語の例から推察できるとおり，個別の分類項目が指す意味の範疇はかなり広い．

分類語彙表は意味を階層的に整理するものの，見出し語は分類項目にのみ属するため，単語間の上位下位関係は示さない．たとえば<哺乳類>の上位範疇は<動物>だが，これに属する見出し語はない．そのため，<哺乳類>に含まれる

見出し語，たとえば**動物**の上位語は特定できない．これは，WordNet の Synset と上位下位関係が形成する概念階層との大きな相違点である．一方で，分類番号の先頭桁が上位の意味的範疇を示すという特性は，本研究で提案する階層コード表現（§ 3.3.1）において先頭の非ゼロ桁が上位語を示すという特性と共通する部分がある．

2.4 単語埋め込みと語彙資源の統合

本研究では，大規模な言語データによって学習した単語埋め込みを，人間が構築した語彙資源の知識に適応させる手法を提案する．その関連研究として，Word2Vec や BERT などの静的または文脈依存の単語埋め込みに対して，WordNet や辞書などの語彙資源から得られる単語の意味や意味関係の知識を統合する事例を紹介する．

これらの事例における方策は 2 つに大別される．ひとつは，語彙資源に含まれる意味関係などを教師信号として，学習ずみの単語埋め込みを更新または変換することにより意味適応させる方策であり，主に静的埋め込みに対して用いられる．なお意味適応とほぼ同義の概念として Refrofitting (Lengerich et al. 2018) があるが，本節では両者を同じものとして扱う．もうひとつは，埋め込み学習における本来の目的関数を拡張して，語彙資源を教師信号とする目的関数をともに最適化する同時学習（joint training）の方策であり，主に文脈依存埋め込みに対して用いられる．本研究で提案するふたつの手法は，いずれも意味適応の手法であるため，前者との関連性が高い．なかでも静的埋め込みを概念階層に適応させる既存研究は，本研究（§ 3）で提案する手法と動機を共有しており，特に関連性が高い．一方で同時学習に属する手法は，埋め込みの計算方法（単語列を文脈依存埋め込みに変換するモデル）を改変するため，本研究との関連性は低い．むしろ原理的には本研究の提案手法（§ 4）と併用可能である．提案手法は BERT が計算した埋め込みへの適用を想定しているが，同時学習を採用した非 BERT モデルが計算した埋め込みにも同じように適用できるためだ．ただし，非 BERT モデルを用いることでタスクの性能が向上するかは明らかではない．

2.4.1 静的埋め込みと語彙資源の統合

静的埋め込みと語彙資源を統合する目的は，語彙資源から得られる意味関係の知識を埋め込みに反映することにより，文脈類似性に基づく意味的類似度と，人間が整理した詳細な意味関係の両方を獲得することである．Word2Vec や fastText が大規模言語データから学習した単語間の類似度は，必ずしも人間の期待どおりになるとは限らないし，対義・上位下位・全体部分などの意味関係の違いは，文脈類似性だけでは学習が難しいためである（§ 2.1.3）．各事例を特徴づける主な項目は，学習対象，学習方法，教師信号である．これらを用いて整理した事例

の一覧を表 2.5に示す.

表 2.5: 静的埋め込みと語彙資源の統合に関する研究の一覧

研究	学習対象	学習方法	教師信号
Retrofitting	意味的関連性	埋め込みを更新	上位下位，同義
Functional Retrofitting	意味的関連性	埋め込みを更新	上位下位，同義，対義など
HyperVec	概念階層	Skip-Gram (§ 2.1) を拡張	上位下位
LEAR	概念階層	埋め込みを更新	上位下位，同義，対義
SPON	概念階層	変換関数を学習	上位下位
DOE	概念階層	ゼロから表現学習	上位下位
Poincaré	概念階層	ゼロから表現学習	上位下位

もっとも基本的な意味適応の方法論は，既存の単語埋め込みを更新または変換して意味的な関連性を反映することである．Retrofitting (Faruqui et al. 2015) および Functional Retrofitting (Lengerich et al. 2018) は，もとの埋め込みからの乖離を抑えつつ，上位下位語対や同義語対を教師信号として互いに近づけるように更新する手法である．これにより，人間が付与した意味的類似性およびトピック的な関連性との相関が高くなることが報告されている．これらの手法は埋め込みを直接更新するため，教師信号に含まれる単語のみが適応の対象となる．Retrofitting で埋め込みの更新に用いられた目的関数を以下に示す.

$$L = \sum_{w \in \mathbb{V}} \alpha_w \|\boldsymbol{v}_w - \hat{\boldsymbol{v}}_w\|^2 + \sum_{(w,w') \in \mathbb{S}} \beta_{w,w'} \|\boldsymbol{v}_w - \boldsymbol{v}_{w'}\|^2 \tag{2.20}$$

$\mathbb{S}$ は教師信号に含まれる単語対，$\hat{\boldsymbol{v}}$ は既存の単語埋め込みである.

上位下位関係識別への応用を想定した意味適応としては，概念階層を擬似的に表現する事例がある．HyperVec (Nguyen et al. 2017) および LEAR (Lexical Entailment Attract-Repel) (Vulic and Mrksic 2018) は，ベクトルの長さ（ノルム）によって概念の抽象度，ベクトルの向き（cosine 類似度）によって上位下位関係とそれ以外の関係を表すことで，単語対の上位下位関係らしさを定量化する関数を導入し，ベクトル空間で階層構造を擬似的に表現することを提案した．LEAR で導入された上位下位関係らしさを計量する関数の一例を以下に示す.

$$l(\boldsymbol{v}_w, \boldsymbol{v}_{w'}) = 1 - \cos(\boldsymbol{v}_w, \boldsymbol{v}_{w'}) + \frac{\|\boldsymbol{v}_w\| - \|\boldsymbol{v}_{w'}\|}{\max\{\|\boldsymbol{v}_w\|, \|\boldsymbol{v}_{w'}\|\}} \tag{2.21}$$

このような関数は厳密に推移律および反対称律を満たすことを志向したものではなく，むしろ図 2.1に示すとおり，概念階層を直感的に模擬する目的で導入されたものである．LEAR はこのような計量を目的関数に組み込んだうえで，上位下位・同義語対は互いに近づけ，対義語対は遠ざけるように埋め込みを更新し，推論時にも同様の計量を用いることで，上位下位関係識別タスクの性能を改善できることを示した．本研究の提案手法（§ 3）の相違点は，単語埋め込みを，包含関係を定義可能な階層コードにもとづく表現形式に変換することである．

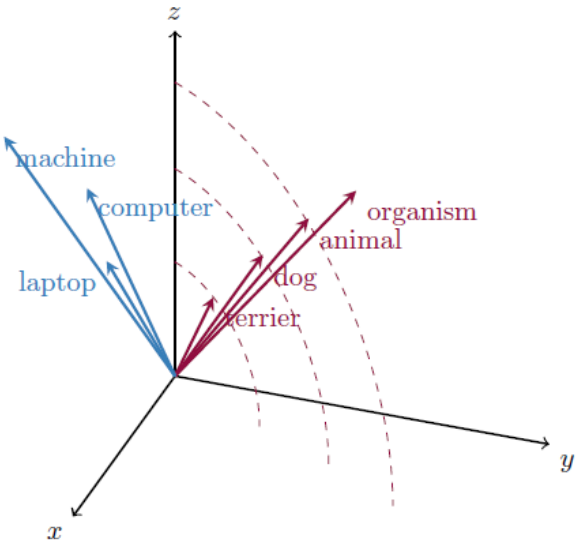


図 2.1: ベクトルの長さと向きで概念階層を擬似的に表現する方法．Vulic and Mrksic (2018) の Figure 1 を引用．

ベクトル空間で概念階層を表現する別の方法論として，SPON (Strict Partial Order Network) (Dash et al. 2020) は，ベクトルの次元ごとに数値の大小を比較することで，上位下位関係らしさを定量化する関数を導入した．

$$l(\boldsymbol{v}_w, \boldsymbol{v}_{w'}) = \sum_{i=1}^{d} \max(0, \epsilon + v_{w,i} - v_{w',i}) \tag{2.22}$$

$v_{,i}$ は i 次元目の値，ϵ はハイパーパラメータである．SPON は上位下位語対を教師信号として，上位下位関係らしさが大きくなるように埋め込みを変換する関数を学習し，全単語を適応できるようにした．また適応ずみ埋め込みを用いて推論することで，上位下位関係識別タスクの性能を改善できることを示した．SPON と本研

究の 3 章で提案する手法は，表現形式および上位下位関係らしさを定量化する方法が異なるが，動機は非常に似ている．なお彼らが用いた語彙資源は，WordNet ではなく，Hearst pattern (Hearst 1992) と呼ばれる，*X is a Y* のような手がかり表現を用いて生コーパスから収集した上位下位語対である (Roller et al. 2018)．彼らの研究は，人手ではなく言語データから自動構築した語彙資源によっても上位下位関係の知識を学習することが可能であることを示唆している．

2.4.2 文脈依存埋め込みと語彙資源の統合

BERT による文脈依存埋め込みと語彙資源を統合する目的は，静的埋め込みと同様に，語義や意味関係を埋め込みに反映することにより，語義の区別や概念の語釈文への対応付けを可能にすることである．主流の方法論は，事前学習の段階で，大規模言語データを教師信号とするマスク言語モデリング（§ 2.2.2）のタスクに加えて，語彙知識を教師信号とする追加タスクと同時に学習させることである．すなわち，BERT を用いて計算した埋め込みをあとから適応させるのではなく，同時学習によって BERT が語彙知識を埋め込みに反映する方法を習得させるという方法論である．もうひとつの統合手法は，意味関係によるネットワークをあらかじめ埋め込みに変換しておき，Transformer の中間層で単語埋め込みと混合できるようにモデルアーキテクチャを拡張することである．すなわち，意味関係を表す特徴量を参照しながら事前学習させることで，やはり BERT が語彙知識を埋め込みに反映する方法を習得させるという方法論である．各事例を特徴づける主な項目は，方法論，学習対象，学習タスクである．これらを用いて整理した事例の一覧を表 2.6に示す．

表 2.6: 文脈依存埋め込みと語彙資源の統合に関する研究の一覧

研究	方法論	学習対象	学習タスク
Levine et al. (2020)	同時学習	語義	Supersense 予測
Chen et al. (2022)	同時学習	単語と語釈文の対応	語釈文から単語を予測
Yu et al. (2022)	同時学習	単語と語釈文の対応	単語から語釈文を識別
Peters et al. (2019)	モデル拡張	意味ネットワーク	Synset 埋め込みと混合
Wang et al. (2021)	意味適応	意味的な関連性	語釈文どうしを近づける

Levine et al. (2020) は，Supersense 予測タスクとの同時学習を提案し，教師あり語義曖昧性解消タスクの性能を改善できることを実証した．Supersense（§ 2.3.1）は 45 種類の高度に抽象化された概念であり，予測タスクでは，文中で

マスクされた単語 w_t が取りうるすべての候補 $A(w_t)$ を正解とみなして学習する．たとえば $w_t = \textit{chocolate}$ ならば $A(w_t) = \{\texttt{noun.food}, \texttt{noun.attribute}\}$ となる．本タスクの目的関数を以下に示す．

$$L = \log \sum_{s \in A(w_t)} p(s|w_1 w_2 ... \texttt{[MASK]} ... w_T) \tag{2.23}$$

Chen et al. (2022) および Yu et al. (2022) は，語釈文から単語（辞書の見出し語）を予測するタスクや，単語と語釈文の埋め込みを互いに近づけるタスクとの同時学習を提案した．後者のタスクでは，見出し語 w_t を含む単語列 $w_1 w_2 ... w_t ... w_T$，見出し語に対応する語釈文 s_P，対応しない語釈文 s_N の 3 つを同時に入力する．そして文中でマスクされた見出し語の埋め込みを，対応する語釈文と近づけて，対応しない語釈文とは遠ざける．たとえば $w_t = \textit{Covid-19}$ ならば $s_P = \textit{Covid-19 is the desease ...}$，$s_N = \textit{SARS a viral respiratory illness ...}$ となる．彼らは GLUE ベンチマーク (Wang et al. 2018) に含まれる多様な評価タスクで性能を計測し，軽微な性能改善を報告している．

Wang et al. (2021) は，意味的に関連する語義の語釈文どうしを近づけるタスクとの同時学習を提案した．このタスクでは，意味関係 r でつながる Synset の対 (X, Y) に対して，語釈文 s_X および s_Y をそれぞれ独立に BERT に入力して，文頭に付与された特殊トークン $w_0 = \texttt{[CLS]}$ の埋め込み $\boldsymbol{v}_{w_0}^{(X,Y)}$ を語釈文の埋め込みとする．語釈文どうしの距離を計量する関数を以下に示す．

$$L = \|\boldsymbol{v}_{w_0}^{(X)} + \boldsymbol{t}_r - \boldsymbol{v}_{w_0}^{(Y)}\|_1 \tag{2.24}$$

ここで $\boldsymbol{t}_r$ は，意味関係 r に固有のベクトルパラメータである．彼らは固有表現どうしの関係を識別するタスクで有効性を評価し，軽微な性能改善を報告している．

Peters et al. (2019) は，WordNet を知識グラフの一種とみなしてあらかじめ埋め込みに変換しておき，中間層で単語埋め込みと混合するモデルアーキテクチャを提案した．具体的には，WordNet の Synset を意味関係でつないだネットワークに対して，知識グラフ埋め込みを学習する手法 (Balazevic et al. 2019) を適用して，Synset の埋め込みを計算する．つぎに Transformer の中間層のアーキテクチャを拡張し，周辺単語の埋め込みと加重平均を取る際（§ 2.2.2）に，単

語がとりうる語義の Synset 埋め込みとも加重平均を取る．この手法は教師あり語義曖昧性解消タスクを用いて有効性が評価され，わずかながら性能の改善が報告されている．

ここまでで説明した既存研究の手法は，BERT の事前学習タスクまたはモデルアーキテクチャを変更することにより，埋め込みに語彙知識を効果的に反映するものである．したがって本研究が提案する，意味ネットワークへの適応による語義曖昧性解消の手法とは原理的に併用可能である．なぜなら，提案手法は BERT が計算した埋め込みをあとから適応させるためである．ただし，BERT と異なる埋め込み計算モデルを使用してタスクの性能が向上するかは明らかでない．実際に各手法を提案した論文での有効性評価では，性能改善は軽微であったり，汎用的な性能改善を目的とした BERT の派生モデル (Liu et al. 2019) に劣る結果が報告されたものが多い．この現象の一因として，マスク言語モデリングに用いる大規模言語データと比較して，語彙資源は桁違いに小規模であることが指摘されている (Berend 2023)．

BERT によって計算した埋め込みをあとから語彙知識に適応させる手法としては，SREF (Wang and Wang 2020) がある．SREF では，WordNet の Sense に対して，Lemma・同義語・語釈文・用例（§ 2.3.1）を連結した文を BERT に入力して，単語ベクトルの算術平均を語義埋め込みとする．つぎに上位下位などの意味的に関連する語義と加重平均を取ることで，互いに近づける．このように適応させた語義埋め込みを用いて，語義曖昧性解消タスクにおいて，対象語の最近傍語義を選ぶことで性能を改善できることが示された．本研究の提案手法（§ 4）の相違点は，語義だけでなく対象語の埋め込みも適応させること，および意味的に関連しない語義や，おなじ単語の異なる語義は遠ざけることである．

第 3 章

概念階層への適応による上位下位関係識別

本章では，静的単語埋め込みを概念の階層構造に適応させる手法の提案および，適応させた表現を用いることで，文脈から独立した単語間の意味関係を問う上位下位関係識別タスクの性能が改善できるかを検証する．

3.1 概要

単語間の上位下位関係は，含意関係認識 (Dagan et al. 2013) およびテキスト生成 (Biran and McKeown 2013) などの幅広いタスクで用いられる．また，上位下位語対の集合を階層構造に集約すると，タクソノミを構築できる (Bordea et al. 2016)．こうした中で上位下位関係識別は，タクソノミの自動構築を実現する基礎技術として位置付けられている (Camacho-Collados 2017)．上位下位関係識別とは，文脈と独立して与えられる単語ペア，たとえば{*animal, dog*}が上位下位関係かそれとも別の意味関係か，上位下位関係ならばどちらが上位語か，どの程度典型的かなどを評価するタスクである．

上位下位関係の背景にある概念階層が持つ特徴は，概念のあいだに包含関係 $\prec$ が存在し，推移律および反対称律が成り立つことである．推移律とは $s \prec t$ かつ $t \prec u$ ならば $s \prec u$ が成立すること，反対称律とは $s \prec t$ ならば $t \nprec s$ が成立することである．これに対して静的単語埋め込みはユークリッド空間上のベクトルであり，包含関係は存在しない．また単語埋め込みは，意味的類似性やトピック的な関連性，および単純な意味関係を捉えているが，同義・対義・上位下位・全体部分などの詳細な意味関係の違いは捉えていない（§ 2.1.3）．

提案手法は，WordNet から抽出した上位下位語ペアを教師信号として，静的

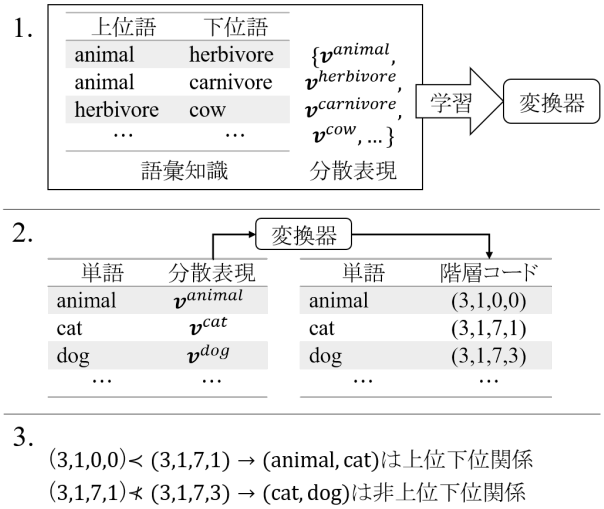


図 3.1: 提案手法全体の模式図．1. 上位下位語ペアと単語埋め込みを用いて変換器を学習， 2. 学習した変換器を用いて埋め込みを階層コード表現に変換， 3. 階層コード間の包含関係を用いて上位下位関係を識別．

単語埋め込みを階層コード表現に変換するモデルを学習する．学習したモデルを用いて埋め込みを変換すると単語に階層コードが割り当てられ，コードの包含関係によって上位下位関係を直接推論できるようになる．階層コードとは，M 進 N 桁，ただしひとたび 0 が出現したら後続桁はすべて 0 となる離散ベクトルであり，先頭の非ゼロ桁数および値の一致によって包含関係を定義する．たとえば $animal$ が cat の上位語であることは，$animal$ は $(3,1,0,0)$，cat は $(3,1,7,1)$ というコードにすることで表現する．課題は，このような都合の良い，たとえば $animal$ が下位概念である dog や cat などを包含するようなコードに変換するモデルを，上位下位語ペアを教師信号として勾配降下法で学習することである．階層コードは離散表現なので，単語ペアの上位下位関係は「あり」か「なし」かの二値になってしまい，勾配情報が得られない．そこで提案手法では，連続緩和したコード表現を用いる．つまり各桁に対して離散値（たとえば 1 桁目に 2）を出力するかわりに連続緩和したコード（たとえば $M = 3$ 進なら $[0.1, 0.0, 0.8]$）を出力し，各桁の値はこれを分布パラメータとするカテゴリカル分布（2 が出る確率が 0.8）に従うものとする．これにより，各単語を表す連続緩和したコード表現から，包含関係を満たすコードのペアが得られる期待値，つまり微分可能な 0 から 1 の連続変数を計算できるようになる．この期待値を，単語ペアが上位下

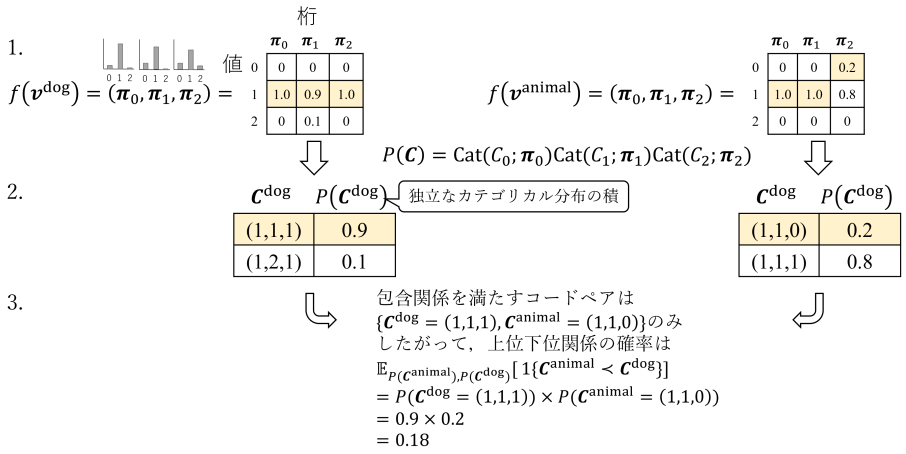


図 3.2: 連続緩和したコード表現（3 進 3 桁）を用いて単語ペアが上位下位関係である確率を計算する模式図．正確な定義は § 3.3.3を参照のこと．1. 変換器（図中 f）が連続緩和したコードを出力． 2. 連続緩和したコードを用いてコードの確率分布を定義（図中 Cat はカテゴリカル分布）． 3. 包含関係を満たすコードペアの確率を計算.

位である確率とする．提案手法全体の模式図を図 3.1に示す．また連続緩和したコード表現を用いて単語ペアが上位下位である確率を計算する模式図を図 3.2に示す．さらに，実際に提案手法により得られた階層コードを表 3.1に示す．ただしわかりやすさのために，連続緩和したコードの各桁で argmax を取って離散化したコード（§ 3.5.4）を掲載した.

表 3.1: 提案手法により得られた，離散化した階層コード（8 進 16 桁）.

単語	階層コード
animal	(2,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0)
mammal	(2,2,2,2,2,2,2,4,0,0,0,0,0,0,0,0)
carnivore	(2,2,2,2,2,2,2,4,0,0,0,0,0,0,0,0)
cat	(2,2,2,2,2,2,2,4,4,5,1,1,7,4,2,2)
dog	(2,2,2,2,2,2,2,4,5,5,1,1,7,4,4,2)
mouse	(2,2,2,2,2,2,2,4,5,5,1,2,7,4,2,2)

提案手法の有効性は，上位下位関係識別の分類タスクおよびランキングタスクによって実験的に検証する．また誤り分析や有効性の要因分析によって，概念の抽象度や意味関係の違いをどの程度捉えられているのかを調べる．また提案手法によって意味の概念階層にどの程度適応したかを調べるため，単語への階層

コードの割り当て方をWordNetの概念階層と比較分析する．具体的には，離散化した階層コードが単語の意味によるクラスタリングであると見立てて，WordNet Synsetの上位下位関係および同義関係が構成する，概念および単語の階層構造と比較する．

3.2 関連研究

本研究の提案手法は，意味適応（Semantic Specialization）およびOrder Embeddings (Vendrov et al. 2016) によるアプローチの長所を統合する手法と位置づけられる．これらのアプローチはいずれも語彙資源を活用した単語埋め込みの表現学習であるが，長所には違いがある．まず上位下位関係識別の精度という観点からは，大規模言語データから学習した意味的類似性が活用できることから，意味適応が有望である (Vulic and Mrksic 2018)．一方でタクソノミ構築を志向する場合には，推論結果は精度だけでなく推移律や反対称律といった望ましい性質を満たすことが重要だと指摘されており (Camacho-Collados 2017)，この観点では順序や包含関係を定義可能なOrder Embeddingsが有望である．こうした中で，両者を兼ね備えた手法はいまだ提案されていない．本研究の提案手法は，単語埋め込みを変換してコード表現を得ることにより意味適応の長所を，包含関係を計量可能な階層コード表現を用いることによりOrder Embeddingsの長所を取り入れるものといえる．また，本研究の提案手法と関連性が深いのはコード学習の手法である．

本研究における上位下位識別タスクの設定は，意味適応やOrder Embeddingsの既存研究と同様に，WordNetのような語彙資源から得られる知識，特に上位下位語ペアを教師信号とする表現学習である．すなわち，既存の単語埋め込みを特徴量とする識別器を教師あり学習で構築するのではなく，獲得した表現を直に用いて上位下位関係を推論するという設定である．

3.2.1 Order Embeddings

Order Embeddings (Vendrov et al. 2016) とは，順序や包含関係を定義可能な表現形式または空間による埋め込み表現の総称である．その長所は，推移律および反対称律という，概念階層が有する性質（§ 2.3.1）が，完全ではなくてもおのずと満たされることである．`DOE` (Density Order Embeddings) (Athiwaratkun and Wilson

2018) は，単語をガウス分布で表現し，上位下位関係らしさを分布の重なり具合で定量化する関数を導入した．Poincaré (Poincaré embeddings) (Nickel and Kiela 2017) は，単語を双曲空間上のベクトルとして表現し，当該空間上で上位下位語対が互いに近くなるように表現学習を行う手法を提案した．

Order Embeddings の短所は，表現学習の対象が語彙資源により制約されることである．各単語の埋め込み表現は語彙資源から得られる上位下位語ペアを用いて学習されるため，語彙資源に含まれない単語については表現が与えられないし，上位下位関係を推論することもできない．これに対して提案手法の相違点は，階層コードをゼロから学習するのではなく，既存の単語埋め込みを変換する方法を学習することである．このため提案手法は語彙資源により制約されず，原理的には埋め込みが所与の単語すべてに表現を与えられる．

3.2.2 意味適応

意味適応とは，語彙資源から得られる単語間の意味関係を既存の埋め込みに反映する手法の総称である（§ 2.4.1）．特に上位下位関係の場合は，ベクトルのノルム長さにより意味階層を，cosine 類似度により関係の強さを表現する方法が提案されている (Nguyen et al. 2017)．LEAR (Lexical Entailment Attract-Repel) (Vulic and Mrksic 2018) はこの表現方法に則って，上位下位・同義・対義関係を反映するように単語埋め込みを更新する手法であり，これまでの最高精度を達成している．これに対して提案手法の相違点は，単語埋め込みを，包含関係を定義可能な階層コードを連続緩和した表現に変換することである．

3.2.3 コード表現

コード学習とは，ユークリッド空間上のベクトルを M 進 N 桁の離散ベクトルに変換する手法である．この手法の長所は，もとの空間上での類似性を維持したまま離散空間上でコンパクトな表現を得ることである．既存研究では，モデル圧縮 (Shu and Nakayama 2018)，クラスタリング (Hu et al. 2017)，および類似文書検索 (Shen et al. 2018, Zheng et al. 2020) に応用されている．一方で，本研究のように包含関係を定義可能なコード表現を学習する手法は提案されていない．

コード先頭の非ゼロ桁によって意味の階層性を表現する方法論は，分類語彙表 (国立国語研究所 2004) の分類番号（§ 2.3.2）のように，古くから知られてい

るものである．一方で，分類語彙表における分類番号の割り当ては意味的範疇の階層構造を陽に把握して行われるのに対して，本研究での単語への階層コードの割り当ては，階層構造内のノードペア，すなわち上位下位語対の集合から変換器を学習することにより暗黙に実行される．

3.3 提案手法

本節では，単語埋め込みを階層コードを連続緩和した表現に変換する変換器を訓練する手法および，コード間の包含関係を計量する手法を提案する．手法の概要は以下の通りである．

1. 単語埋め込みを変換器に入力して，階層コードの確率分布を出力する．
2. Gumbel-Softmax Trick を用いて，確率分布からサンプリングする．これにより，各桁の one-hot vector を連続緩和した形式で，階層コードのサンプルを得る．
3. 得られたサンプルを用いて，上位下位識別，再構築損失，および相互情報量からなる目的関数を計算する．上位下位識別は，上位下位語ペアを正例，非上位下位語ペアを負例とする二値分類である．
4. 目的関数に対するモデルパラメータの勾配を計算する．
5. 勾配を用いて，変換器のモデルパラメータを更新する．

手法の特徴は，本来の問題を連続緩和して扱うことである．本来の階層コードは，各桁が離散値で表される．しかし離散値のままでは，勾配降下法による最適化が困難である．そこで変換器の訓練および上位下位関係の推論を行うための計算過程では，各桁の one-hot vector を連続緩和した形式を用いて計算を行う．つまり本来の階層コードは確率変数として扱うことにして，変換器は連続緩和したコードを出力し，それが階層コードの確率分布を規定する．また階層コードの各桁は互いに独立で，連続緩和したコードを分布パラメータとするカテゴリカル分布としてモデル化する．これにより，単語どうしが上位下位関係になる確率を，それぞれの確率分布から包含関係を満たすような階層コードのペアが得られる期待値として，しかも解析解を計算することが可能になる．

以下，それぞれの計算を詳述する．

3.3.1 階層コードおよび上位下位関係の定義

本論文の階層コードの定義は M 進 N 桁，ただしひとたび 0 が出現したら後続桁はすべて 0 となる離散ベクトルである．たとえば $(3, 1, 0, 0)$ は階層コードである．深さ N，最大幅 M の木構造は，幅優先符号化により階層コードに変換できる．ここから直感的にわかるように，階層コードを用いて包含関係を定義できる．すなわち a) 上位より下位のほうが非ゼロ桁数が多く，かつ b) 双方ともに非ゼロのすべての桁について，値が一致する 関係であると定義すればよい．たとえば $((3, 1, 0, 0), (3, 1, 7, 0))$ は包含関係である．

3.3.2 変換器

変換器（エンコーダ）は，単語埋め込み $\boldsymbol{v}$ を階層コード $\boldsymbol{C} = (C_0, C_1, ..., C_{N-1})$ の確率分布 $P(\boldsymbol{C})$，厳密にはその分布パラメータに変換する関数である．階層コードで包含関係を表すためには，上位桁の値が下位桁の選択に影響を及ぼす必要がある．そこで LSTM による再帰的計算および，訓練時は Gumbel-Softmax Trick (Maddison et al. 2017) によるサンプリングを用いて，下位桁の確率分布が上位桁の値に依存する関数をモデリングする．なお後者の仕掛けは確率質量を特定の値に偏らせて one-hot vector に近付ける役割もある．アーキテクチャの模式図を図 3.3に示す.

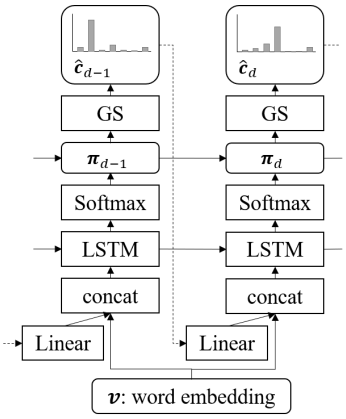


図 3.3: 変換器のアーキテクチャ．角丸は変数，直角は関数，点線は誤差逆伝搬しないことを表す．記号の定義は本文を参照.

d桁目の階層コードを表す確率変数およびその値を C_d および $a \in \{0, 1, ..., M-1\}$ とする．ひとたび 0 が出現したら後続桁はすべて 0 となるという制約より，上

位 $d-1$ 桁 $\boldsymbol{C}_{<d}$ を所与とした C_d の条件付分布は

$$\begin{aligned} & P(C_d = a|\boldsymbol{C}_{<d}) \\ = \; & \mathbb{1}_{\{a=0\}} P(C_{d-1} = 0|\boldsymbol{C}_{<d-1}) \\ & + P(C_d = a|C_{d-1} \neq 0, \boldsymbol{C}_{<d}) P(C_{d-1} \neq 0|\boldsymbol{C}_{<d-1}) \end{aligned} \tag{3.1}$$

と表される．ここで $\mathbb{1}_{\{.\}}$ は指示関数を表す．また右辺第 1 項および第 2 項はそれぞれ，直前桁がゼロの場合および非ゼロの場合に対応している．これにより，下位桁になるにつれて値がゼロになる確率が大きくなることが保証される．

直前桁が非ゼロの場合の確率分布 $P(C_d = a|C_{d-1} \neq 0, \boldsymbol{C}_{<d})$ のカテゴリカル分布パラメータ $\boldsymbol{\pi}'_d \in \mathbb{\Delta}^{M-1}$（ただし $\mathbb{\Delta}^{M-1}$ は $M-1$ 次元の単体を表す[1]）は，LSTM を用いてモデリングする．LSTM の入力は，直前桁の値および，変換対象の単語埋め込みである．

$$\boldsymbol{\pi}'_d = \mathrm{Softmax}(\mathrm{Linear}(\boldsymbol{h}_d)) \tag{3.2}$$

$$\boldsymbol{h}_d = \mathrm{LSTM}([\boldsymbol{v}; \mathrm{Linear}(\mathrm{detach}(\hat{\boldsymbol{c}}_{d-1}))], \boldsymbol{h}_{d-1}) \tag{3.3}$$

ここで $\boldsymbol{v}$ は単語埋め込み，$\boldsymbol{h}_d$ は d 回目の LSTM の隠れ状態ベクトル，; はベクトルの連結，Linear は線形変換層を表す．ただし線形変換層への入力については誤差逆伝搬をしない[2]（式中 detach）．

$\hat{\boldsymbol{c}}_{d-1} \in \mathbb{\Delta}^{M-1}$ は，$d-1$ 桁目の値の one-hot vector を連続緩和したものである．すなわち $\hat{\boldsymbol{c}}_{d-1}$ は M 次元ベクトルであり，a 番目の要素は $d-1$ 桁目が a を取る割合を示している．したがって $0 \leq \hat{c}_{d-1,a} \leq 1$ かつ $\sum_{a=0}^{M-1} \hat{c}_{d-1,a} = 1$ を満たす．

$\hat{\boldsymbol{c}}_{d-1}$ の計算は，訓練時は Gumbel-Softmax Trick（式中 GS）によるサンプリング，推論時はカテゴリカル分布パラメータをそのまま用いる．

$$\hat{\boldsymbol{c}}_{d-1} = \begin{cases} \mathrm{GS}(\boldsymbol{\pi}_{d-1}) & \text{訓練時} \\ \boldsymbol{\pi}_{d-1} & \text{推論時} \end{cases} \tag{3.4}$$

[1] $M-1$ 次元の単体とは，すべての要素がゼロ以上かつ，かつ要素の合計が 1 をとる M 次元の実数ベクトルの集合である．Softmax 関数の出力，カテゴリカル分布のパラメータ，および Gumbel-Softmax Trick の出力はすべて，単体上の点である．

[2] 予備実験において，誤差逆伝搬を有効にすると学習が不安定になることを確認したため．なお上位桁への誤差逆伝搬は $\boldsymbol{h}_d$ および $\boldsymbol{\pi}'_d$ を経由する．

最後に，式 3.1に従って $\boldsymbol{\pi}'_d$ を補正し，条件付確率 $P(C_d = a|C_{<d})$ のカテゴリカル分布パラメータ $\boldsymbol{\pi}_d \in \mathbb{\Delta}^{M-1}$ を求める．

$$\pi_{d,0} = \pi_{d-1,0} + (1 - \pi_{d-1,0})\pi'_{d,0} \tag{3.5}$$

$$\boldsymbol{\pi}_{d,>0} = \frac{(1 - \pi_{d,0})}{(1 - \pi'_{d,0})}\boldsymbol{\pi}'_{d,>0} \tag{3.6}$$

ベクトルの添字 $d, > 0$ は，先頭の次元を除くことを表す．

以上により，階層コードの確率分布 $P(\boldsymbol{C})$ は

$$P(\boldsymbol{C}) = \prod_{d=0}^{N-1} \mathrm{Cat}(C_d; \boldsymbol{\pi}_d) \tag{3.7}$$

と定義される．ただし $\mathrm{Cat}(;\boldsymbol{\pi})$ は，$\boldsymbol{\pi}$ をパラメータとするカテゴリカル分布である．また $\boldsymbol{\pi}$ は連続緩和したコードでもある．なお訓練時には Gumbel-Softmax Trick を適用するため，$\boldsymbol{\pi}$ は確率的であることに注意されたい．推論時は確定的である．

3.3.3 上位下位関係の計量

上位下位関係の計量は，単語埋め込み $(\boldsymbol{v}^s, \boldsymbol{v}^t)$ をそれぞれ変換した確率分布から得られる階層コード $(\boldsymbol{C}^s, \boldsymbol{C}^t)$ を用いる．ただし s および t はそれぞれ，上位語候補および下位語候補を表す．いま，階層コードの観測値が得られたとしよう．ふたつの階層コードを比較すると，その関係はかならず，包含・一致・その他のいずれかに分類できる．包含関係の定義をアルゴリズムの形式で書き下したものが，Algortithm 1に示す判定関数 RELATION である．たとえば包含 $(= 1)$ を返す If ブロックの条件式：$C^s_d = 0 \wedge C^t_d \neq 0$ は，上位下位関係の定義（§ 3.3.1）で述べた条件 “a) 上位より下位のほうが非ゼロ桁数が多く” に対応している．

判定関数 RELATION を用いて，任意の単語ペアに対して上位下位関係の計量ができる．具体的には，s が t の上位語である関係を（§ 2.2と同様に）$t \to s$ と書くことにすると，上位下位関係の確率 $P(t \to s)$ は RELATION が包含 $(= 1)$ を返す期待値となる．同様に，上位下位関係でない確率 $P(t \nrightarrow s)$ は，RELATION

Algorithm 1 階層コードの関係を判定する関数

Require: $(\boldsymbol{C}^s, \boldsymbol{C}^t)$: 候補 s および候補 t の階層コード
Output: コードペアの関係．1: 包含, 0: 一致, -1: その他
  function RELATION$(\boldsymbol{C}^s, \boldsymbol{C}^t)$
    for $d = 0$ to $N-1$ **do**
      if $C_d^s = 0 \wedge C_d^t \neq 0$ **then**
        return 1
      else if $C_d^s = C_d^t \wedge C_d^s \neq 0 \wedge C_d^t \neq 0$ **then**
        continue
      else if $C_d^s = 0 \wedge C_d^t = 0$ **then**
        return 0
      else
        return -1
      end if
    end for
    return 0
  end function

がその他 $(= -1)$ を返す期待値となる．

$$P(t \to s) = \mathop{\mathbb{E}}_{P(\boldsymbol{C}^s), P(\boldsymbol{C}^t)} [\mathrm{RELATION}(\boldsymbol{C}^s, \boldsymbol{C}^t) = 1] \tag{3.8}$$

$$P(t \nrightarrow s) = \mathop{\mathbb{E}}_{P(\boldsymbol{C}^s), P(\boldsymbol{C}^t)} [\mathrm{RELATION}(\boldsymbol{C}^s, \boldsymbol{C}^t) = -1] \tag{3.9}$$

ただし $P(\boldsymbol{C}^s)$ および $P(\boldsymbol{C}^t)$ は，式 3.7により定義された，上位語候補 s および下位語候補 t の階層コードが従う確率分布である．

実際には，訓練時には $P(\boldsymbol{C})$ の分布パラメータ $\{\boldsymbol{\pi}_d\}_{d=0}^{N-1}$ は確率的であるため，上述の期待値は解析解を計算できない．かといって，階層コードの実現値をサンプリングしてモンテカルロ近似する方法も，計算コストや勾配の推定精度に問題がある．そこで Variational AutoEncoder (Kingma and Welling 2014) の方法論に倣って，式 3.4で得られるサンプル $\hat{\boldsymbol{C}} = (\hat{\boldsymbol{c}}_0, \hat{\boldsymbol{c}}_1, ..., \hat{\boldsymbol{c}}_{N-1})$ を用いて近似計算する．具体的には，各桁が独立かつ，$\hat{\boldsymbol{c}}_d$ をパラメータとするカテゴリカル分布 $\mathrm{Cat}(C_d; \hat{\boldsymbol{c}}_d)$ を用いて

$$\hat{P}(\boldsymbol{C}) = \prod_{d=0}^{N-1} \mathrm{Cat}(C_d; \hat{\boldsymbol{c}}_d) \tag{3.10}$$

と近似する．そのうえで，式 3.8の $P(\boldsymbol{C}^s)$ および $P(\boldsymbol{C}^t)$ をそれぞれ $\hat{P}(\boldsymbol{C}^s)$ および $\hat{P}(\boldsymbol{C}^t)$ で置き換えて，期待値を計算する．

$$P(t \to s) \approx \mathop{\mathbb{E}}_{\hat{P}(\boldsymbol{C}^s),\hat{P}(\boldsymbol{C}^t)} [\mathrm{RELATION}(\boldsymbol{C}^s, \boldsymbol{C}^t) = 1] \tag{3.11}$$

$$P(t \nrightarrow s) \approx \mathop{\mathbb{E}}_{\hat{P}(\boldsymbol{C}^s),\hat{P}(\boldsymbol{C}^t)} [\mathrm{RELATION}(\boldsymbol{C}^s, \boldsymbol{C}^t) = -1] \tag{3.12}$$

$P(\boldsymbol{C})$ を $\hat{P}(\boldsymbol{C})$ で置き換えて期待値を計算することは，P の分布パラメータを 1 個の Gumbel-Softmax Trick のサンプルで近似していることになる．

訓練時の $\hat{P}(\boldsymbol{C})$ は各桁を独立な分布で近似，また推論時の $P(\boldsymbol{C})$ は Gumbel-Softmax Trick によるサンプリングがない（式 3.4）ことから，やはり各桁は独立である．したがっていずれの場合も判定関数 RELATION の期待値は，モンテカルロ近似に頼らず解析解が計算できる．これを，単語ペアの上位下位関係 $P(t \to s)$ および，非上位下位関係 $P(t \nrightarrow s)$ の確率とする．以下に，解析解の計算方法を説明する．なお定式化には訓練時の分布パラメータ $\hat{\boldsymbol{c}}_d$ を用いるが，推論時は $\boldsymbol{\pi}_d$ に読み替えればよい．

まず，計算方法の基本的な考え方を説明する．$\hat{P}(\boldsymbol{C})$ は各桁が互いに独立なので，d 桁目の値が a を取る確率は $P(C_d = a) = \hat{c}_{d,a}$ となることを思い出そう．これを利用すると，RELATION の If ブロック条件式が成立する割合（確率）が求められる．たとえば条件式：$C^s_d = 0 \wedge C^t_d \neq 0$ が成立する割合は $\hat{c}^s_{d,0}(\sum_{a\neq 0} \hat{c}^t_{d,a})$ と計算できる．さらに If 文は上位桁から順に反復する For ループで囲まれているため，RELATION が $\{-1, 0, 1\}$ を返す割合は，d 桁目ではじめて該当する If ブロックの条件式が成立する割合を，すべての桁について合計した値とすればよい．

定式化を続ける前に，各桁を互いに独立としても階層コードの定義と矛盾しないことを説明する．たしかに各桁が互いに独立の確率分布は，階層コードの定義を満たさない事象，たとえば $(3, 1, 0, 1)$ のように後続桁で非ゼロがふたたび出現する事象の確率がゼロにならない．すなわち $\hat{P}(\boldsymbol{C})$ は，厳密には階層コードのみを台とする確率分布ではない．しかし上位下位関係の計量においては，以後の定式化でも明らかなとおり，ゼロが出現して For ループが終了する桁以降は計量に寄与しない．つまり階層コードの定義を満たす事象のみが上位下位関係の計算対象になるため，階層コードの定義との矛盾はない．

さて，前述した考え方にもとづき，まずは上位下位関係の確率 $P(t \to s)$ すなわち RELATION が 1 を返す場合を定式化する．$\beta_d^{st}, \gamma_d^{st}, \delta_d^{st}$ を，それぞれ 1 を返す If ブロック，For ループ継続の If ブロック，0 を返す If ブロックの条件式が成立する割合

$$\beta_d^{st} := P(C_d^s = 0 \wedge C_d^t \neq 0) \tag{3.13}$$

$$\gamma_d^{st} := P(C_d^s = C_d^t \wedge C_d^s \neq 0 \wedge C_d^t \neq 0) \tag{3.14}$$

$$\delta_d^{st} := P(C_d^s = 0 \wedge C_d^t = 0) \tag{3.15}$$

と定義すると，これらの変数の値は

$$\beta_d^{st} = \hat{c}_{d,0}^s(1 - \hat{c}_{d,0}^t) \tag{3.16}$$

$$\gamma_d^{st} = \sum_{a=1}^{M-1} \hat{c}_{d,a}^s \hat{c}_{d,a}^t \tag{3.17}$$

$$\delta_d^{st} = \hat{c}_{d,0}^s \hat{c}_{d,0}^t \tag{3.18}$$

と計算できる．なお γ_d^{st} の計算式における添字 a が 1 から始まる理由は，条件式：$C_d^s \neq 0 \wedge C_d^t \neq 0$ を満たすには $a = 0$ を取りえないためである．これを $d = 0, 1, ...N-1$ 桁目まで合計することにより，$P(t \to s)$ は

$$P(t \to s) = \sum_{d=0}^{N-1} \beta_d^{st} \left(\prod_{d'=-1}^{d-1} \gamma_{d'}^{st} \right);\ \gamma_{-1}^{st} = 1 \tag{3.19}$$

と計算できる．

同様に $P(t = s)$ すなわち RELATION が 0 を返す場合については

$$P(t = s) = \prod_{d=0}^{N-1} \gamma_d^{st} + \sum_{d=0}^{N-1} \delta_d^{st} \left(\prod_{d'=-1}^{d-1} \gamma_{d'}^{st} \right) \tag{3.20}$$

と計算できる．

これらを用いて，非上位下位関係の確率 $P(t \nrightarrow s)$ は

$$P(t \nrightarrow s) = 1 - (P(t \to s) + P(t = s)) \tag{3.21}$$

と計算できる．

3.3.4 目的関数

変換器（エンコーダ）を最適化する際の目的関数は，上位下位関係の識別，再構築損失，非ゼロ桁数と単語埋め込みの相互情報量の重み付き和とする．原理的には上位下位関係の識別のみを最適化すればよいが，再構築損失および相互情報量を補助的に併用することで，コードの偏りを防ぐとともに，埋め込み空間上の類似性が階層コードに反映されやすくする (Shu and Nakayama 2018, Hu et al. 2017)．

$$L = L_{\mathrm{h}} + \alpha L_{\mathrm{reconst}} + \beta L_{\mathrm{mi}} \tag{3.22}$$

上位下位関係の識別に対する目的関数 L_{h} は，上位下位語ペアを正例（$y = 1$），非上位下位語ペアを負例（$y = 0$）とする二値分類に対するクロスエントロピー誤差として定義する．

$$L_{\mathrm{h}} = \sum_{(s,t,y)\in\mathbb{H}^+\cup\mathbb{H}^-} y \ln P(t \to s) + (1-y) \ln P(t \nrightarrow s) \tag{3.23}$$

ただし $\mathbb{H}^+ = \{(s,t,y=1)\}$ は，語彙資源から抽出した上位下位語ペアの集合である．同様に $\mathbb{H}^- = \{(s,t,y=0)\}$ は，$\mathbb{H}^+$ から機械的に生成する非上位下位語ペアの集合である．生成方法については§ 3.3.5で述べる．

再構築損失に対する目的関数 L_{reconst} は，階層コードから再構築した埋め込みと，オリジナルの埋め込みとの L2 誤差として定義する．ただし再構築には非ゼロ桁のみが寄与するように $a > 1$ のみを参照する．具体的には

$$L_{\mathrm{reconst}} = \sum_{w\in\mathbb{V}} ||\hat{\boldsymbol{v}}^w - \boldsymbol{v}^w|| \tag{3.24}$$

$$\hat{\boldsymbol{v}} = \frac{||\boldsymbol{v}||}{||\hat{\boldsymbol{v}}'||}\hat{\boldsymbol{v}}',\ \hat{\boldsymbol{v}}' = \sum_{d=0}^{N-1}\sum_{a=1}^{M-1} \hat{c}'_{d,a}\boldsymbol{e}_{d,a} \tag{3.25}$$

$$\hat{c}'_{d,a} = \hat{c}_{d,a}\left(\prod_{k=0}^{d-1}(1-\hat{c}_{k,0})\right) \tag{3.26}$$

と定義する[3]．ただし $\mathbb{V}$ は，単語埋め込みの語彙である．また $\boldsymbol{e}_{d,a}$ は，d 桁目の値 $a(\neq 0)$ に割り当てた基底ベクトルである．基底ベクトルは他のモデルパラメータと同様に，訓練時に最適化する．

階層コードの非ゼロ桁数と単語埋め込みの相互情報量に対する目的関数 L_{mi} は，以下の式で計算できる．

$$
\begin{aligned}
L_{\mathrm{mi}} &= -\mathrm{I}(\mathrm{length}(\boldsymbol{C}); V) && (3.27)\\
P(V=\boldsymbol{v}) &\approx \frac{1}{|\mathbb{V}|}\sum_{w\in\mathbb{V}} \mathbb{1}\{\boldsymbol{v}=\boldsymbol{v}^w\} && (3.28)
\end{aligned}
$$

ただし length(.) は，階層コードの非ゼロ桁数を返す関数である．つまり length($\boldsymbol{C}$) は，$\{0,1,...,N\}$ のいずれかを取る確率変数である．また V は，単語埋め込みを表す確率変数である．

相互情報量 I(.) を求めるためには，非ゼロ桁数および単語埋め込みそれぞれの確率分布が必要である．そこで，単語埋め込みの確率分布は経験分布で近似する．また非ゼロ桁数の確率分布 $P(\mathrm{length}(\boldsymbol{C})=n|V)$ は，上位下位関係の計量と同様に，各桁を独立と近似した階層コードの確率分布 $\hat{P}(\boldsymbol{C})$（式 3.10）から解析的に導出する．そのうえで，相互情報量の計算式を説明する．

まず，確率分布を用いた記法に書き直して見通しをよくしよう．また記述を簡略化するために $\mathrm{length}(\boldsymbol{C})=S$ と表記する．相互情報量，エントロピー，周辺化分布の定義より

$$
\begin{aligned}
\mathrm{I}(S;V) &= \mathrm{H}(S)-\mathrm{H}(S|V) && (3.29)\\
\mathrm{H}(S) &= -\sum_{n=0}^{N} P(S=n)\ln P(S=n) && (3.30)\\
\mathrm{H}(S|V) &= \mathop{\mathbb{E}}_{P(V)}\left[\sum_{n=0}^{N} P(S=n|V)\ln P(S=n|V)\right] && (3.31)\\
P(S=n) &= \mathop{\mathbb{E}}_{P(V)}\left[P(S=n|V)\right] && (3.32)
\end{aligned}
$$

となる．すなわち $\mathrm{I}(S;V)$ は，非ゼロ桁数の確率分布について，無条件分布と単語埋め込みによる条件付分布それぞれのエントロピーの差である．

つぎに，非ゼロ桁数の単語埋め込みによる条件付確率分布を定義する．非ゼ

[3]上付き添字 w が明らかな場合は添字を省略した．

ロ桁数 $n \in \{0, 1, ..., N\}$ の定義は，はじめて値がゼロになる桁位置である（最後までゼロが出現しなければ $n = N$ になる）．近似した階層コードの確率分布 $\hat{P}(\boldsymbol{C})$ を用いると，d 桁目の値がゼロになる確率は $\hat{c}_{d,0}$ なので

$$\begin{aligned} P(S = n|V = \boldsymbol{v}) &\approx P(S = n|\hat{\boldsymbol{C}}) \\ &= \hat{c}_{n,0} \prod_{d=-1}^{n-1} (1 - \hat{c}_{d,0}) \\ \text{ただし} \quad & \hat{c}_{-1,0} = 0,\ \hat{c}_{N,0} = 1 \end{aligned} \tag{3.33}$$

と計算できる．

最後に，単語埋め込みの確率分布を経験分布で近似する．

$$P(V = \boldsymbol{v}) \approx \frac{1}{|\mathbb{V}|} \sum_{w \in \mathbb{V}} \mathbb{1}\{\boldsymbol{v} = \boldsymbol{v}^w\} \tag{3.34}$$

以上を組み合わせると，式 3.29で定義した無条件分布および条件付分布のエントロピーは

$$\mathrm{H}(S) = -\sum_{n=0}^{N} P(S = n) \ln P(S = n) \tag{3.35}$$

$$\mathrm{H}(S|V) \approx \frac{1}{|\mathbb{V}|} \sum_{w \in \mathbb{V}} \sum_{n=0}^{N} P(S = n|V = \boldsymbol{v}^w) \ln P(S = n|V = \boldsymbol{v}^w) \tag{3.36}$$

$$P(S = n) \approx \frac{1}{|\mathbb{V}|} \sum_{w \in \mathbb{V}} P(S = n|V = \boldsymbol{v}^w) \tag{3.37}$$

と計算できる．

3.3.5 非上位下位語ペアの生成

上位下位関係の識別に対する目的関数 L_{h} を最適化するためには，正例と負例，すなわち上位下位語ペアと非上位下位語ペアが必要である．そこで順序反転および乱択を用いて，個別の上位下位語ペアから 5 つの非上位下位語ペアを生成する．順序反転は単語順序の交換，乱択は片方の単語をランダムサンプリングした単語と交換する操作である．乱択は，30%の確率で単語埋め込みの最近傍 100 語から，70%の確率で全語彙からサンプリングする．ただし，乱択した単語が元

の単語と上位下位関係になる場合は棄却する．これにより，関係性に乏しい単語ペアだけでなく，意味的類似性は高いが上位下位関係ではない単語ペアも生成されるようにする．非上位下位語ペアの生成例を表 3.2に示す．

表 3.2: 非上位下位語ペアの生成．イタリック体は乱択された単語を表す．

操作	下位語 (t)	上位語 (s)
正例	dog	animal
順序反転	animal	dog
上位語を最近傍から乱択	dog	*dog food*
上位語乱択+順序反転	*dog food*	dog
下位語を全語彙から乱択	*captain*	animal
下位語乱択+順序反転	animal	*captain*

3.4 実験結果

3.4.1 評価タスク・データセット・推論方法

表 3.3: 上位下位識別評価タスクの一覧

種類	タスク名	事例数	事例	正解
分類	BLESS-hyponymy	1,337	{*snake*, *creature*}	*creature*
分類	WBLESS	1,668	*tail* → *fox*	`false`
分類	BIBLESS	1,668	*appliance* → *stove*	下位上位
ランキング	HyperLex	2,616	*soda* → *liquid*	138 位

先行研究 (Nguyen et al. 2017, Vulic and Mrksic 2018) に倣い，分類タスク 3 種類および，ランキングタスク 1 種類を用いて，上位下位識別タスクにおける提案手法の性能を評価する．評価タスクの概要を表 3.3に示す．

分類タスクは3種類である．具体的にはBLESS-hyponymy (Kiela et al. 2015), WBLESS (Weeds' BLESS) (Weeds et al. 2014)，BIBLESS (Kiela et al. 2015) を用いる．BLESS-hyponymy は上位下位語ペアのうちどちらが上位語かの 2 値分類，WBLESS は上位下位関係・その他の 2 値分類，BIBLESS は上位下位関係・下位上位関係・その他の 3 値分類を解くタスクである．評価データセットはいずれも BLESS（§ 2.3.2， Baroni and Lenci (2011)）のサブセットから作成されて

いる．BLESS-hyponymy は上位語とのペア，WBLESS および BIBLESS は上位下位・下位上位・全体部分・無関係の 4 種類の単語ペアで構成されている．

推論は，上位下位関係にある確率 $P(t \to s)$（式 3.8）を用いて行う．判定規則は以下のとおりである．

- BLESS-hyponymy は事例 $\{t, s\}$ が与えられる．$P(t \to s)$ と $P(s \to t)$ をそれぞれ計算して，$P(t \to s) < P(s \to t)$ ならば t，そうでなければ s が上位語だと判定する．
- WBLESS は事例 $t \to s$ が与えられる．$P(t \to s)$ がしきい値以上ならば上位下位関係，そうでなければ上位下位関係でないと判定する．
- BIBLESS は事例 $t \to s$ が与えられる．まず $\max\{P(t \to s), P(s \to t)\}$ をしきい値と比較する．しきい値未満の場合は，その他と判定する．しきい値以上の場合は $P(s \to t) < P(t \to s)$ ならば上位下位関係，そうでなければ下位上位関係と判定する．

WBLESS および BIBLESS の判定で用いるしきい値は，開発データに最適化する．また開発データ・テストデータ分割は先行研究 (Nguyen et al. 2017) に倣って，データセットからそれぞれ 2%，98%にランダム分割する．評価指標はテストデータの正解率（accuracy）である．ただしデータセット分割におけるランダムネスの影響を排除するため，開発データ・テストデータ分割を 1,000 回実施して平均値を求める．

ランキングタスクは，HyperLex (Vulic et al. 2017) を用いる．HyperLex は与えられた単語ペアを上位下位関係らしさの高い順に順位付けして，アノテータが付与した順位との一致度を評価するタスクである．評価データセットは HyperLex（§ 2.3.2）である．推論は $P(t \to s)$ の値が大きい順に順位付けする．評価指標はスピアマンの順位相関である．

3.4.2 学習方法

単語埋め込みおよび語彙資源

提案手法で用いる静的単語埋め込みおよび語彙資源は，先行研究 (Nguyen et al. 2017) に倣う．具体的には，静的単語埋め込みは fastText (Bojanowski et al. 2017)

（§ 2.1）を用いる．また上位下位識別 L_{h} の最適化に用いる上位下位語ペアは，WordNet の上位下位関係を用いる．

fastText は，Mikolov et al. (2018) が配布するサブワード情報付きモデル[4] を用いる．これにより，未知語を生じさせずにすべての単語を埋め込みに変換できる．ベクトルの次元数は 300 である．語彙数は 100 万で，大文字・小文字を区別する．また *dog food* のような複単語表現の場合は，全単語の算術平均を取る．

上位下位語ペアの抽出は，直接または間接に Hypernym の意味関係にある Sense ペアの Lemma を用いる．たとえば *cat* → *feline*（直接の Hypernym），*cat* → *mammal*，*cat* → *carnivore*（間接の Hypernym）などの単語ペアが得られる．そのうえで，評価タスクのデータセットである BLESS および HyperLex と重複する単語ペアを，順序を入れ替えたペアも含めて削除する．したがって（当然ではあるが），推論時の上位下位関係識別は訓練データには出現しない単語ペアに対して行うことになる．サンプル数は，名詞ペアが 2,158,824 件，動詞ペアが 162,706 件となった．

なお訓練データに含まれる上位下位語ペアを推移律でつなぐだけでは，必ずしも正しい上位下位関係を復元できないことに注意されたい．具体例のひとつは，評価データとの重複の削除や，そもそも語彙資源が網羅していないという理由で連鎖が途切れる場合である．たとえば評価データが想定する正しい Hypernym の連鎖は $W \to X \to Y \to Z$ だが，訓練データには $X \to Y$ がないとしよう．このとき上位下位語ペア $\{W \to X, W \to Y, W \to Z, X \to Z, Y \to Z\}$ をつなぐだけでは元の連鎖は復元できず，$X \to Y$ は確定しない．逆に多義語の影響により，上位下位語ペアをつなぐことで誤った上位下位関係を推論する場合もある．たとえば *mouse* には “動物のねずみ” と “臆病な人” の語義があり，前者は *wood mouse* を下位語に，後者は *person* を上位語に取る．したがって語義が複数あることを考慮せず上位下位語ペアをつなぐと *wood mouse*→*person* を上位下位関係だと誤って推論してしまう．これらの例から，上位下位関係を正しく推論するためには上位下位語ペア以外にも情報や仕掛けが必要であることがわかる．

[4] https://fasttext.cc/docs/en/english-vectors.html

最適化およびハイパーパラメータ

目的関数の最適化は，ミニバッチによる確率勾配法を用いる．上位下位識別 L_{h} のミニバッチサンプル数は，正例 200 件・負例 1,000 件である．再構築損失 L_{reconst} および相互情報量 L_{mi} のミニバッチサンプル数は 1,000 件である．最適化アルゴリズムは Sharpness-aware Minimization Optimizer (Foret et al. 2020) を用いる．Gumbel-Softmax Trick の温度パラメータは 1.0 とする．これは PyTorch package:`gumbel_softmax` 関数のデフォルト値である．なお予備実験では，温度パラメータの影響は些少であった．目的関数の重み付き和（式 3.22）は $\alpha = 5.0, \beta = 0.05$ とする．階層コードの基数 M および桁数 N は，8 進 16 桁とする．

3.4.3 結果

表 3.4: 提案手法の性能および先行研究との比較．提案手法は 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を上回る場合は有意差（スチューデントの両側 t 検定[5]，$*: p < 0.05$）を報告．太字は各タスクの最高精度．

手法	語彙資源	意味関係	B-hyp	WB	BIB	HLex
`Poincaré` (Nickel and Kiela)	WordNet	上位下位	-	0.860	-	0.512
`DOE` (Athiwaratkun and Wilson)	WordNet	上位下位	-	-	-	0.590
`SPON` (Dash et al.)	Hearst ptn.	上位下位	0.970	0.910	0.870	-
`HyperVec` (Nguyen et al.)	WordNet	上位下位	0.920	0.870	0.810	0.54
`LEAR` (Vulic and Mrksic)	WordNet, Roget	上位下位，同義，対義	0.960	**0.920**	0.880	**0.686**
提案手法	WordNet	上位下位	**0.984*** (0.004)	0.919 (0.004)	**0.886** (0.007)	0.539 (0.012)

表 3.4に，提案手法の性能および先行研究との比較を示す．提案手法は，分類タスクにおいて高い性能を示した．特に BLESS-hyponymy (B-hyp) および BIBLESS (BIB) では，`SPON` (Dash et al. 2020) および `LEAR` (Vulic and Mrksic 2018) による先行研究の最高精度を，それぞれ 1.4 ポイントおよび 0.6 ポイント上回った．また，訓練に使用された語彙資源および意味関係の違いを考慮に入

[5]Wilcoxon の符号順位検定（両側検定）も実施した．p 値は，BLESS-hyponymy が 0.0625（$2/2^5 = 0.0625$ より，サンプル数 5 における理論的な最小値），BIBLESS が 0.1875 であり，スチューデントの両側 t 検定の結果とほぼ同じだった．

れても，提案手法は先行研究に対して優位性のある結果になっている．具体的には，本研究とほぼ同様の設定である HyperVec (Nguyen et al. 2017) と比較すると，提案手法は 5 から 7 ポイントの性能改善を示している．また本研究と異なり同義・対義関係も用いている LEAR と比較しても，提案手法は同程度あるいは 1 から 2 ポイントの性能改善を示している．これらの結果は，提案手法は上位下位関係に関する語彙知識の活用効率が高いことを示唆している．

ランキングタスクである HyperLex (HLex) においては，提案手法は LEAR による最高精度を 10 ポイント以上下回った．この要因としては，最適化に用いる目的関数の影響が考えられる．提案手法はクロスエントロピー誤差を最小化するため，上位下位語ペアは 1，非上位下位語ペアは 0 に近づけようとする．このため，上位下位関係らしさのスコア $P(t \rightarrow s)$ は二極化しやすい．これに対して LEAR はヒンジ損失の最小化を用いる（すなわち，マージン以下の誤差は許容される）ため，二極化は起きにくいと考えられる．また前述したとおり，LEAR は本研究よりも多くの語彙知識を用いていることにも留意が必要である．

3.5 分析

3.5.1 分類タスクの誤り分析

表 3.5: 非上位下位語ペアの意味関係別正答率

意味関係	WBLESS		BIBLESS	
	正解	正答率	正解	正答率
上位下位	true	0.901	上位下位	0.893
下位上位	false	0.976	下位上位	0.907
同位	false	0.914	その他	0.873
全体部分	false	0.838	その他	0.843
部分全体	false	0.966	その他	0.805
無関連	false	0.930	その他	0.915
すべて	-	0.919	-	0.886

WBLESS および BIBLESS で用いられる非上位下位語ペアは，同位や全体部分などの意味関係から採録されている．したがって，非上位下位語ペアのみに着目して意味関係ごとに正答率を算出することで，どのような意味関係を上位下位関係と誤認しやすいのかを把握できる．意味関係ごとの正解率を表 3.5に示

す．WBLESS は 2 値分類（上位下位か否か，つまり `true` か `false`），BIBLESS は 3 値分類（上位下位，非上位下位，その他）であることに注意されたい．

提案手法は，BIBLESS の全体部分関係および部分全体関係における正答率が相対的に低いことがわかる．すなわち提案手法は，{*fox,mouth*}や{*radio,wire*}のような単語ペアを，上位下位関係または下位上位関係だと誤認しやすいのである．なお WBLESS の部分全体関係は正答率が高いことから，部分語が上位語候補にくる場合（例：*fox*→*mouth*）は誤認しにくいことがわかる．したがって，全体語および部分語をそれぞれ概念階層の上位（非ゼロ桁数が小）および下位（非ゼロ桁数が大）だと捉えることはできているが，上位下位関係ではなく全体部分関係だということは捉えられていないといえる．この要因としては，訓練時に生成する非上位下位語ペアに全体部分関係が少ないことが影響していると考えられる．提案手法は，語彙全体または意味的類似度が高い単語群からの乱択によって非上位下位語ペアを生成する．このため非上位下位語ペアは，意味関係がないか，または意味的類似性が高いかのいずれかになりやすいが，全体部分関係は必ずしもどちらにも該当しないためである．逆に言えば，部分全体語ペアは訓練データに含まれていないにもかかわらず，階層コードは概念階層上の深さ（概念の抽象度）に適応できていることが示唆される．

3.5.2　ランキングタスクの誤り分析

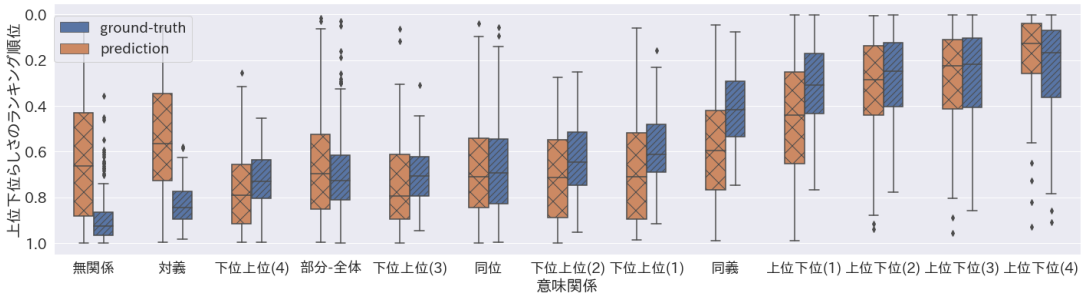


図 3.4: HyperLex データセットの意味関係ごとの順位分布．順位は全サンプル数で正規化．意味関係のカッコ内の数値は上位語・下位語間のホップ数を表す

HyperLex データセットは，同義や対義など，上位下位関係ではない意味関係からも単語ペアを採録している．また，単語ペアに付与されている上位下位関係らしさのスコアは，意味関係ごとに水準が異なると報告されている (Vulic et al.

2017)．HyperLex タスクは上位下位関係らしさを順位付けするので，意味関係ごとに予測した順位と正解の順位を比較すれば，どのような意味関係で上位下位関係らしさを過大または過小に評価する傾向があるのかを把握できる．この発想にもとづき，意味関係ごとに順位の分布を可視化した結果を図 3.4に示す．提案手法は，上位下位・同義・下位上位については，平均的には正解と同様の順位付けができているが，しばしば正解よりもばらつきが大きいことがわかる．また対義関係では正解より高い順位を予測している，つまり上位下位らしさを過大評価していることがわかる．この要因としては，非上位下位語ペアに対義関係が出現しにくいことが考えられる．理由は分類タスクの誤り分析で述べたものとおなじである．したがって Vulic and Mrksic (2018) に倣って対義関係を負例に用いることで，性能を改善する余地があるかもしれない．

3.5.3 有効性の要因

表 3.6: 提案手法と，手法の一部を無効化した場合との比較．アブレーション列は無効化した対象を示す．角カッコ内の数値は提案手法との accuracy の差（単位はポイント）．寄与度は accuracy 差の全タスク平均値.

アブレーション	B-hyp	WBLESS	BIBLESS	HyperLex	寄与度
提案手法	0.984	0.919	0.886	0.539	-
– 目的関数; 再構築損失	0.988 [+0.3]	0.918 [−0.1]	0.870 [−1.5]	0.556 [+1.7]	+0.1
– 目的関数; 相互情報量	0.980 [−0.5]	0.921 [+0.2]	0.880 [−0.5]	0.539 [−0.0]	−0.2
– 負例生成; 最近傍から乱択	0.979 [−0.6]	0.913 [−0.6]	0.869 [−1.7]	0.530 [−0.9]	−0.9

表 3.7: BIBLESS タスクにおける意味関係ごとのアブレーション結果.

意味関係	– 再構築損失 [pt]	– 最近傍から乱択 [pt]
上位下位	0.7	0.2
下位上位	0.5	−0.5
同位	−7.9	−12.4
全体部分	−8.0	−1.1
部分全体	−4.9	−1.8
無関連	−1.9	−0.2
すべて	−1.5	−1.7

提案手法では，階層コードへの変換器を学習するための手段として，補助目的関数および非上位下位語ペアの自動生成を用いている．これらの手段が上位下位識別タスクの性能に寄与しているかどうかを調べるために，アブレーションによる有効性分析を実施する．提案手法と，手段の一部を無効化した場合との性能差を表 3.6に示す．この結果から，最近傍からの乱択による非上位下位語ペアの生成が，上位下位識別タスク全般に有効であることがわかる．また BIBLESS に限定すれば，階層コードから単語埋め込みを再構築する目的関数（再構築損失）が有効だとわかる．最近傍からの乱択および再構築損失は，意味的類似度が高い単語に対して異なるコードの割り当てを促す効果がある．したがって，上位下位関係か否かの識別が促進されるのだと考えられる．より詳細に調べるため，BIBLESS タスクにおいて意味関係ごとにアブレーションした結果を表 3.7に示す．再構築損失または最近傍からの乱択を無効化すると，同位関係および全体部分関係の性能が 8 から 12 ポイント低下する一方で，無関連な単語ペアでの性能差は 2 ポイント未満だとわかる．したがって前述の考察のとおり，これらの手段は意味的類似度が高い単語ペアに対して上位下位関係か否かを学習する効果が高いことが示唆される．

再構築損失が有効であることは，Order Embeddings の実現方法として階層コード表現を用いることの長所ともいえるだろう．再構築損失は Denoising AutoEncoder の枠組みで導入できるため，変換器（エンコーダ）・逆変換器（デコーダ）アーキテクチャの設計が必要である．コード表現の場合は，Shu and Nakayama (2018) や本研究の提案手法（式 3.24）のように，各桁の値に固有の基底ベクトルを導入することで容易に逆変換器が設計できる．一方で，Order Embeddings の既存研究では確率分布 (Athiwaratkun and Wilson 2018) や超直方体 (Li et al. 2019) などを用いる手法が提案されているが，これらの表現からベクトルに戻す逆変換器の設計は，コード表現の場合ほど直接的ではない．

3.5.4 階層コードの割り当て特性の分析

提案手法は，単語埋め込みを M 進 N 桁の階層コードに変換する．実際には各桁は連続緩和されている（§ 3.3）が，仮に各桁で最大値を取る要素を選択すれば，単語を M^N 通りのいずれかに割り当てる，クラスタリングの一種とみなせる (Hu et al. 2017)．一方で WordNet についても，Synset（§ 2.3.1）を基準とす

れば，同義関係によるクラスタリングとみなせる．そこで本節では，提案手法による階層コードをクラスタリング手法とみなす場合に，どのような割り当て特性を持つのかを分析する．具体的には WordNet の Synset によるクラスタリングとの比較および，階層コードの基数・桁数による影響を調べる．

分析方法

クラスタリングの対象とする語彙は，本実験と同様に，WordNet から抽出した上位下位語ペアの集合（§ 3.4.2）に出現するすべての単語とする．ただし WordNet の構造と基準を合わせるため，異なる品詞は異なる単語として扱う．たとえば名詞の *play* と動詞の *play* は異なる単語とする．語彙数は 116,510 件で，そのうち名詞が 105,048 件，動詞が 11,462 件である．

階層コードによるクラスタの割り当ては，各桁で最大値を取る要素を選択する．すなわちクラスタ ID を $z_0 z_1 ... z_d ... z_{N-1}$ として

$$z_d = \mathrm{argmax}_a(\pi_{d,a} | a \in \{0, 1, ..., M-1\}) \tag{3.38}$$

と割り当てる．なお $\boldsymbol{\pi}_d$ は，d 桁目の値が従うカテゴリカル分布のパラメータである（式 3.7）．また単語埋め込みは表層形ごとに固有なので，品詞による差異はない．WordNet の Synset によるクラスタの割り当ては，Synset ID をクラスタ ID とする．ただし多義語の場合は，使用頻度がもっとも高いと期待される WordNet first sense（§ 2.3.1）が属する Synset を選択する．

割り当て特性の評価指標は，クラスタの統計量，クラスタリングの精度，およびクラスタ内・クラスタ間の単語類似度の 3 種類を用いる．クラスタの統計量は，クラスタ数およびクラスタ別の平均単語数を用いる．クラスタリングの精度は，Synset が形成する同義語クラスタを正解として，Adjusted Rand Index (ARI) (Hubert and Arabie 1985) で定量化する．ARI は部分集合群への割り当てかたの類似性を表す指標であり，でたらめな割り当てならば 0，完全一致ならば 1 となる．

単語類似度は，クラスタ内の場合は同一クラスタから，クラスタ間の場合は異なるクラスタから，それぞれ同一品詞の単語ペアを乱択して評価する．サンプル数は 10,000 件である．類似度指標は，WordNet の上位下位関係による概念階層における類似度および，単語埋め込みの類似度の 2 種類を用いる．計量指

標はそれぞれ，Wu-Palmar (WuP) 類似度および，cosine 類似度を用いる．WuP は木構造におけるノード間の近さを 0 から 1 で表す指標であり，一致ならば 1 となる (Wu and Palmer 1994)．なお本節での略称はそれぞれ，概念階層的類似度および埋め込み類似度とする．

モデル訓練時の統計誤差の影響を考慮するため，階層コードによるクラスタの割り当ては，モデルパラメータの初期値を変えて 5 回試行する．

WordNet の概念階層との比較

階層コード学習に用いる語彙知識は，WordNet から抽出した上位下位語ペアである．したがって，階層コードの割り当て方は WordNet の概念階層と関連するふるまいが期待される．実用的にも，もしも階層上の位置が近い単語に同一のコードが割り当てられる傾向が確認できれば，同義関係や同位関係の識別可能性に対する示唆にもなる．こうした動機から，階層コードによるクラスタリングの特性を，Synset によるクラスタリングの特性と比較することで，WordNet の上位下位関係および同義関係が構成する概念および単語の階層構造との関連性を分析する．比較結果の一覧を，表 3.8に示す．またクラスタ内単語数のヒストグラムを図 3.5に，同一クラスタ内または異クラスタ間での単語類似度の分布を図 3.6に示す．

表 3.8: クラスタによる割り当て特性の比較．階層コードは初期パラメータを変えて 5 回実験した平均および標準偏差（カッコ内の数値）を報告．

クラスタ	クラスタ数	平均単語数	ARI	概念階層的類似度		埋め込み類似度	
				クラスタ内	クラスタ間	クラスタ内	クラスタ間
Synset	70,959	1.64	-	1.00	0.23	0.62	0.32
階層コード	71,564	1.64	0.0024	0.53	0.23	0.73	0.32
	(7,120)	(0.15)	(0.0011)	(0.11)	(0.00)	(0.11)	(0.01)

クラスタ数および平均単語数は，階層コードの特性が WordNet Synset の特性とよく一致した．たとえば平均単語数は，階層コードと WordNet Synset のいずれも 1.64 件となった．一方でヒストグラムの比較からは，階層コードは，一部のクラスタに割り当てが集中する傾向が強いことがわかった．たとえば 15 単語以上が属するクラスタ数は，階層コードでは 284 個あるが，WordNet Synset では 5 個のみであった．また階層コードによるクラスタ数の標準偏差は平均の約

10%であり，試行（すなわち，変換器の最適化）によるばらつきがやや大きいことを示唆している．

WordNet Synset との割り当ての一致度を表す指標である ARI は，0.0024 であった．この数値は，でたらめな割り当てよりも明らかによい[6] が，完全一致には程遠いという水準である．すなわち，割り当てた階層コードの同一性を根拠として同義関係を推論することは，きわめて困難であるといえる．

単語類似度は，クラスタ内の概念階層的類似度を除き，階層コードの特性が WordNet Synset の特性とよく一致した．また概念階層的類似度と埋め込み類似度のいずれについても，クラスタ間よりもクラスタ内の類似度のほうが高い．すなわち，同一の階層コードが割り当てられる単語群は，異なる階層コードが割り当てられる単語群と比べて，階層構造上の配置と単語埋め込みがともに似通るといえる．したがって，割り当てた階層コードが一致するか否かは，単語間の同義関係や同位関係を識別する手がかりになりうると示唆される．ただし ARI の水準がほぼゼロであることおよび，概念階層的類似度が 0.53 にとどまっており，正確な同義関係ならば 1.0 になることを考慮すると，改善の余地は非常に大きいといえる．

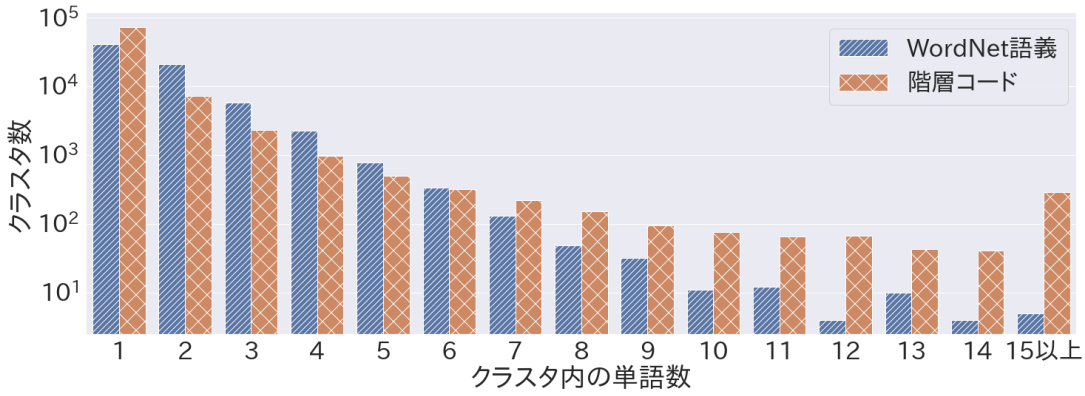


図 3.5: クラスタ内単語数のヒストグラム．

同一の階層コードが割り当てられる単語群の実例を，表 3.9に示す．ひとつめの特徴として，複合語では一部の単語が共通することが挙げられる．たとえば `Noun-d` の単語群は *system* が共通している．この要因としては，複合語の埋め込みは各単語の算術平均を取る（§ 3.4.2）ため，埋め込みが似やすいことが

[6]でたらめな割り当てによる ARI のシミュレーション値は 2.6×10^{-7} であった．

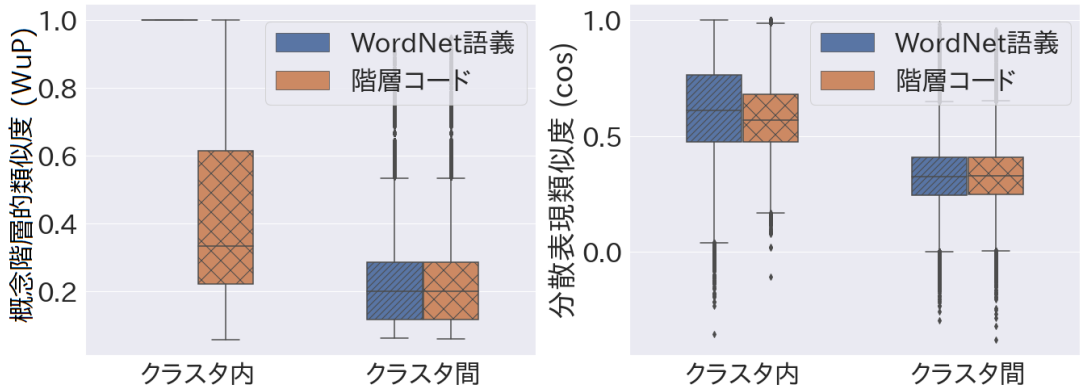


図 3.6: 単語類似度の分布．左：概念階層的類似度 (Wu-Palmar)，右：埋め込み類似度 (cosine)．

挙げられる．ふたつめの特徴として，同位関係が散見されることが挙げられる．たとえば `Noun-b` 単語群の (*Rodentia, Cricetus*) は間接の同位関係 (WuP=0.625)，`Noun-c` 単語群の (*genus Tympanuchus, genus Pyrocephalus*) は直接の同位関係 (WuP=0.875) である．すべての単語が互いに同位関係ではないものの，クラスタ間よりもクラスタ内の概念階層的類似度が高いという結果と整合的である．

表 3.9: 同一の階層コードが割り当てられた単語の例．単語数が $\{2, 4, 6, 8\}$ 個のクラスタを無作為に抽出．ID は便宜的に付与．

ID	単語数	単語群
`Noun-a`	2	*goldfinch, limpkin*
`Noun-b`	4	*Rodentia, Cricetus, Choloepus, Sarcocephalus*
`Noun-c`	6	*genus Tympanuchus, genus Pyrocephalus, genus Rhyncostylis, genus Streptococcus, genus Eucinostomus, genus Stenopterygius*
`Noun-d`	8	*ship-towed long-range acoustic detection system, concentration, localisation, genetic marker, feasibility, naval tactical data system, fixed-point representation system, positional representation system*
`Verb-a`	2	*come upon, look upon*
`Verb-b`	4	*cause, raiment, be active, evict*
`Verb-c`	6	*solemnize, compartmentalise, analogize, allegorize, sermonize, literalize*
`Verb-d`	8	*derogate, enounce, posit, indite, reveal, declaim, attaint, outguess*

基数および桁数の影響

階層コードがとりうる場合の数は，M 進 N 桁のとき，M^N 通りである．したがって基数および桁数を変化させると，クラスタの割り当て特性が変化することが

表 3.10: 基数および桁数の影響．アスタリスク (*) はデフォルト設定との差が統計的に有意（Welch の両側 t 検定，$p < 0.05$）．クラスタは “C” と略記．単語数はクラスタ別平均.

設定	C 数	単語数	ARI	WuP 類似度		cos 類似度		評価タスク			
				C 内	C 間	C 内	C 間	B-hyp	WB	BIB	HLex
提案手法 8 進 16 桁	71,564	1.64	0.0024	0.53	0.23	0.73	0.32	0.984	0.919	0.886	0.539
桁数 2 倍 $N = 32$	103,914*	1.17*	0.0008*	0.48	0.24	0.69	0.33	0.984	0.916	0.873	0.519
基数 2 倍 $M = 16$	101,835*	1.15*	0.0036	0.60	0.24	0.83	0.33	0.985	0.924	0.889	0.540
桁数半減 $N = 8$	6,150*	19.43*	0.0007*	0.42	0.23	0.53*	0.32	0.984	0.925*	0.887	0.564*
基数半減 $M = 4$	19,333*	6.15*	0.0009*	0.41	0.23	0.54*	0.31	0.980*	0.921	0.880	0.538

予想される．こうした動機から，16 進 8 桁のデフォルト設定をベースラインとして，基数および桁数をそれぞれ 2 倍または半減させた場合の割り当て特性を計測し，ベースラインとの差および，差の統計的な有意性を評価する．また参考として，各評価タスクの精度についても評価する．実験結果を表 3.10に示す.

クラスタ数は，基数・桁数の増減と連動して増減する結果となった．提案手法の目的関数は，再構築損失および相互情報量を補助的に併用しているため，とりうる場合の数に連動して，コードの分散・集中を促す度合いが上下動するのだと考えられる.

単語類似度は，基数を 2 倍にすると，統計的に有意ではないものの，クラスタ内の類似度が上昇した．すなわち，基数を増やすと，同一の階層コードが割り当てられる単語どうしの類似性が高まる傾向があるようだ．また ARI も，0.1 ポイントと微量ながら上昇している．したがって，基数を増やすと階層コードの割り当て特性が WordNet の概念階層に近づく傾向があると示唆される.

上位下位識別タスクの精度は，HyperLex (HLex) の一部を除き，ベースラインとの差は 1 ポイント以内であった．またベースラインとの統計的な有意差があるのは 16 通りのうち 3 通りのみであった．したがって基数および桁数は，本タスクに強く影響するパラメータではないといえる.

以上の知見にもとづき，適切な基数および桁数を探索する方法を考察する．仮に上位下位識別タスクの精度のみが関心事の場合は，基数・桁数の影響は小さいため，幅広く探索する意義は小さいといえる．一方で階層コードの割り当て特

性を WordNet の概念階層に近づけることが関心事の場合は，基数および桁数を小さな値から始めて徐々に大きくしてゆき，クラスタ内単語ペアの Wu-Palmar 類似度が低下に転じるか飽和する周辺で探索を止めるのがよいだろう．

3.5.5 間接的な上位下位語ペアの影響

階層コードへの変換器（エンコーダ）の訓練に用いる上位下位語ペアは，直接・間接いずれの上位下位関係でもかまわない．たとえば *cat* → *feline* → *carnivore* という Hypernym の連鎖から，*cat* → *feline* と *cat* → *carnivore* の 2 つを取り出して訓練事例にできる．このため本実験（§ 3.4.3）では，直接（ホップ数 1）か間接（ホップ数 2 以上）かを問わず，Hypernym の連鎖をたどってすべての単語ペアを使用するという設定（§ 3.4.2）での実験結果を報告した．一方で，提案手法は包含関係を定義可能な階層コードという表現を用いるため，原理的には，直接の上位下位語ペアのみで訓練して完全に正確なコードへの変換方法を学習すれば，間接的な上位下位関係を推論できる[7]．したがって，訓練データに含める間接的な上位下位語ペアを意図的に制限する場合に，上位下位識別タスクの性能がどのような影響を受けるのかは，興味深い問いである．仮に間接的な上位下位語ペアを制限しても十分な精度が得られるならば，訓練に要する計算負荷の削減や，上位下位語ペアの網羅性に対する要求の緩和に寄与するためである．特に後者の観点は，使用を想定する語彙資源が，人間向けの国語辞典または平文コーパスから収集した上位下位語ペア (Hearst 1992, Roller et al. 2018, 隅田他 2009) である場合に重要である．これらの語彙資源は，一般に WordNet に比肩するような網羅性は持たないためである．したがって本節では，訓練データに含める間接的な上位下位語ペアをホップ数で制限する場合の影響を評価する．具体的には，すべての単語ペアを用いる無制限の場合をベースラインとして，ホップ数が上限値以下の上位下位語ペアのみを用いる場合のタスク性能を計測し，ベースラインとの精度の差および，差の統計的な有意性を評価する．なおモデル訓練時の統計誤差の影響を考慮するため，各設定に対してモデルパラメータの初期値を変えて 5 回試行する．実験結果を表 3.11に示す．

[7]非上位下位関係については必ずしもそうではない．なぜならば，訓練時に非上位下位語ペアを生成する際に，本来は間接的な上位下位関係にある単語ペアが誤って生成されうるためである（正例になければ棄却できない）．本節の実験ではこの問題を捨象しているため，やや厳しい評価であることに留意されたい．

表 3.11: 間接的な上位下位語ペアをホップ数で制限した場合の影響．角カッコ内の数値はベースライン（無制限）との accuracy の差（単位はポイント）．アスタリスク (*) はベースラインとの差が統計的に有意（Welch の両側 t 検定，$p < 0.05$）．

ホップ数	割合 [%]	B-hyp	WBLESS	BIBLESS	HyperLex
無制限	100.0	0.984	0.919	0.886	0.539
1	14.2	0.952 [−3.3]*	0.808 [−11.1]*	0.710 [−17.6]*	0.447 [−9.2]*
≤ 2	27.9	0.980 [−0.5]	0.858 [−6.0]*	0.783 [−10.2]*	0.501 [−3.7]*
≤ 3	41.8	0.983 [−0.2]	0.893 [−2.6]*	0.838 [−4.8]*	0.526 [−1.3]
≤ 4	54.7	0.983 [−0.1]	0.904 [−1.5]*	0.857 [−2.9]*	0.537 [−0.2]
≤ 5	66.9	0.984 [−0.0]	0.910 [−0.9]	0.868 [−1.8]	0.532 [−0.7]
≤ 7	85.8	0.983 [−0.1]	0.917 [−0.1]	0.879 [−0.7]	0.536 [−0.3]
≤ 9	94.9	0.982 [−0.2]	0.923 [+0.4]	0.886 [−0.0]	0.546 [+0.7]

ホップ数の上限を大きくするにつれて，すなわち概念抽象度の差が大きい上位下位語ペアを訓練データに含めるにつれて，すべてのタスクで精度が向上した．精度向上の傾向はほぼ単調増加で，ベースラインに漸近する傾向を示した．また 5・7・9 ホップ以下の設定ではベースラインとの統計的な有意差が認められないが，精度向上の傾向は飽和していない．したがって，上位下位識別タスクの性能を高くするためには，間接的な上位下位語ペアを網羅的に訓練データに含めること，すなわち仮に直接の上位下位語ペアのみが与えられた場合は，有向非巡回グラフを構築して，間接的な単語ペアを網羅的に生成することが望ましいといえる．

タスクごとにホップ数の影響を比較すると，精度向上幅がもっとも小さいタスクは BLESS-hyponymy (B-hyp) で，もっとも大きいタスクは BIBLESS である．この結果から得られる示唆は，ホップ数が大きい上位下位語ペアは，概念階層の深さを表すコードの非ゼロ桁数を学習することにはさほど貢献しないが，概念階層の包含関係を表す，非ゼロ桁の値を学習することに貢献する，ということである．この示唆は，前者より後者の方が学習の難易度が高いので，より多くのデータを必要とするのだという直感と整合的である．なぜならば，非ゼロ桁数ははじめてゼロを出力する桁を N 通りの選択肢から選ぶのに対して，階層コードの値は各桁ごとに M 通りの選択肢があるからだ．

3.5.6 非上位下位語ペア生成の影響

提案手法に含まれる各種のハイパーパラメータは，上位下位識別タスクの性能に影響すると考えられる．また有効性の要因分析（§ 3.5.3）では，最近傍からの

表 3.12: 最近傍からの乱択に関するハイパーパラメータの影響．角カッコ内の数値はデフォルト設定との accuracy の差（単位はポイント）．アスタリスク (*) はベースラインとの差が統計的に有意（Welch の両側 t 検定，$p < 0.05$）．

設定	B-hyp	WBLESS	BIBLESS	HyperLex
デフォルト $(q, k) = (30\%, 100)$	0.984	0.919	0.886	0.539
最近傍から乱択する確率 (q) を変更				
$q = 0\%$	0.979 [−0.6]	0.913 [−0.6]*	0.869 [−1.7]*	0.530 [−0.9]
$q = 50\%$	0.983 [−0.1]	0.918 [−0.1]	0.880 [−0.6]	0.525 [−1.4]
$q = 70\%$	0.984 [−0.1]	0.910 [−0.9]	0.871 [−1.5]	0.531 [−0.8]
$q = 100\%$	0.985 [+0.1]	0.904 [−1.5]*	0.857 [−2.9]*	0.501 [−3.8]*
最近傍の単語数 (k) を変更				
$k = 10$	0.988 [+0.3]	0.922 [+0.3]	0.888 [+0.2]	0.532 [−0.7]
$k = 30$	0.983 [−0.2]	0.922 [+0.3]	0.888 [+0.2]	0.535 [−0.4]
$k = 300$	0.986 [+0.1]	0.922 [+0.3]	0.890 [+0.4]	0.541 [+0.2]
$k = 500$	0.981 [−0.3]	0.919 [−0.0]	0.876 [−0.9]	0.540 [+0.1]

乱択による非上位下位語ペア（負例）の生成がタスクに有効とする考察を得た．したがって本節では，最近傍からの乱択に関するハイパーパラメータの影響を評価する．具体的には，デフォルトの設定をベースラインとして，最近傍から乱択する確率 q および最近傍の単語数 k を変更した設定におけるタスクの性能を計測し，ベースラインとの精度の差および，差の統計的な有意性を評価する．なおモデル訓練時の統計誤差の影響を考慮するため，各設定に対してモデルパラメータの初期値を変えて 5 回試行する．実験結果を表 3.12に示す．

最近傍の単語数は，すべての設定においてベースラインとの精度差は統計的に有意ではなかった．したがって，提案手法は最近傍の単語数に対して鈍感だといえる．

最近傍から乱択する確率は，0%（最近傍からの乱択が無効）および 100%（全語彙からの乱択が無効）の場合に，デフォルト設定を統計的に有意に下回った．特に 100%の場合は，デフォルト設定を 1.5 ポイントから 3.8 ポイント下回り，大きく精度が低下した．確率を 50%および 70%に増やす場合は，統計的に有意ではないものの，デフォルト設定を 0.1 ポイントから 1.5 ポイント下回った．以上の結果から，最近傍から乱択する確率は，上位下位識別タスクの性能に相応に影響するといえる．最近傍からの乱択を完全に有効化あるいは無効化すると，いずれも精度が低下する．また最近傍から乱択する確率の最適値は，デフォルト設定のとおり，30%前後であると示唆される．

3.5.7　多義語における上位下位関係の分析

階層コードへの変換器（エンコーダ）の訓練に用いる上位下位語ペア $t \to s$ は，多義語に関しては各語義ごとに異なる単語ペアが与えられる．たとえば下位語 t が *mouse* となるペアは，“動物のマウス” の語義からは $mouse \to gnawer$，“臆病な人” の語義からは $mouse \to person$ や $mouse \to somebody$ が与えられる．すなわち下位語が多義語の場合，正解となる上位語が複数与えられる．しかし提案手法は静的埋め込みを採用しているため，各単語にはひとつの階層コードのみが割り当てられる．したがって，すべての正解の上位語に対して上位下位関係の確率 $P(t \to s)$ を同時に最大化するような割り当てはできないと思われる．あるいは，静的埋め込みではひとつのベクトルに複数の意味が混在する（Meaning conflation deficiency, § 2.1.3）という指摘をふまえると，特定の上位語との確率が突出するのではなく，複数の上位語に対して偏りなく高い確率を取る可能性もある．そこで本節では，多義語が下位語である上位下位語ペアについて，複数存在する正解の上位語の中でどの単語が選好されるのかを，上位下位関係の確率 $P(t \to s)$ にもとづき分析する．

分析方法

分析に用いる上位下位語ペアは，本実験で上位下位識別の最適化に用いた訓練データ（§ 3.4.2）のサブセットである．具体的には，直接の Hypernym の意味関係にあり，なおかつ下位語が多義，つまり下位語の Lemma が複数の Sense に紐付く上位下位語ペアを用いる．直接の Hypernym に限定する理由は，間接の Hypernym を除外することで，頻出する（抽象度の高い）*object* のような上位語の影響を抑えるためである．サンプル数は，名詞ペアが 215,312 件，動詞ペアが 66,023 件となった．下位語の異なり数は，名詞が 59,516 件，動詞が 9,482 件となった．これらの単語ペアについて，本実験における推論方法と同様に，上位下位関係にある確率 $P(t \to s)$（式 3.8）を計算した．

具体例の観察

多義の下位語 t の各語義における上位語 s との上位下位関係の確率の実例を，表 3.13に示す．

表 3.13: 下位語 t の各語義における上位語 s との上位下位関係の確率 $P(t \to s)$. 確率の降順に上位 3 件を表示．語義に対して複数の上位語がある場合は，確率が最大の単語を記載．下位語のカッコ内の数字は語義数.

品詞	下位語 (t)	下位語の語義	上位語 (s)	$P(t \to s)$
名詞	*mouse* (4)	“臆病な人”	*soul*	0.72
		“機器のマウス”	*electronic device*	0.06
		“動物のマウス”	*gnawer*	0.02
	arm (6)	“武器”	*instrument*	0.50
		“裾”	*cloth covering*	0.33
		“機械のアーム”	*projection*	0.13
	pupil (3)	“学生”	*enrollee*	0.65
		“生徒”	*young person*	0.31
		“瞳孔”	*aperture*	0.02
動詞	*justify* (5)	“余白を調節する”	*adjust*	0.36
		“正当化する”	*maintain*	0.35
		“言い訳する”	*defend*	0.25
	draft (3)	“草稿を書く”	*compose*	0.39
		“徴兵する”	*enter*	0.37
		“青写真を描く”	*plan*	0.29
	embark (3)	“危険を承知で進める”	*go*	0.41
		“着手する”	*begin*	0.29
		“搭乗する”	*get on*	0.24

これらの実例からは，上位下位関係の確率を説明する一貫した傾向は見出しにくい．たとえば，*mouse* のように特定の語義における上位語との確率が突出する事例と，*draft* のように複数の上位語に対して同程度の確率を取る事例の両方がみられる．また *justify* の “余白を調節する” および “正当化する” のように，明確に異なる語義でも同程度の確率を取る事例もある．語義の典型性や使用頻度が確率に反映されているとも言い難い．*mouse* では “臆病な人” の語義における上位語 *soul* との確率が突出しているが，これが “動物のマウス” や “機器のマウス” より上位語らしいと感じられる (Vulic et al. 2017)，いわゆる典型性が高い単語とは考えにくい.

因子との相関分析

具体例の観察からは，上位下位関係の確率を一貫して説明する規則は見いだせなかった．しかし原理的には，単語に階層コードを割り当てる過程が確率に影響するはずである．そこで本節では，単語埋め込み，および階層コードへの変換器の学習に関わる因子を定量的に定義して，それらの因子と確率の相関を分析す

る．これにより，複数の上位語の中からどの単語が選好されるのかを決めている規則を考察する．

上位下位関係の確率との相関を分析する因子として，上位語の頻度，単語埋め込みの類似度，および下位語の Sense の順位の 3 つを定義する．

- 上位語の頻度は，本実験で用いた上位下位語ペアの訓練データ（§ 3.4.2）の中で，単語 s が上位語として出現した回数である．たとえば *soul* は 11,584 回，*electronic device* は 116 回である．この因子と相関が高い場合は，多数の下位語を持つ上位語が選好されることが示唆される．

- 単語埋め込みの類似度は，下位語と上位語の埋め込みの cosine 類似度である．たとえば $(mouse, soul)$ は 0.31 である．この因子と相関が高い場合は，下位語の埋め込みと類似する上位語が選好されることが示唆される．

- 下位語の Sense の順位は，下位語の Lemma に紐付くすべての Sense の中での順位である．たとえば *mouse* の “臆病な人” は 3 番目である．Lemma に紐付く Sense はおおむね出現頻度順に整列されている（§ 2.3.1）ため，順位が高いほど典型的な語義だとみなす．この因子と相関が高い場合は，下位語の典型的な語義に対応する上位語が選好されることが示唆される．

上位下位関係の確率と因子との関連性は，スピアマンの順位相関によって定量化する．また関連性の可視化のため，各因子の水準を 4 区分に分け，区分ごとに確率の分布を箱ひげ図で可視化する．区分の分けかたは，上位語の頻度および単語埋め込みの類似度については四分位数で分割，下位語の Sense の順位については 1，2，3 番および 4 番以下で分割する．

分析結果を図 3.7に示す．埋め込みの類似度，および下位語の Sense の順位はほぼ相関しないのに対して，上位語の頻度とは明らかな正の相関があることがわかる．これは，多義語への階層コードの割り当てにおいて，多数の下位語を持つ上位語との確率を大きくする選好が働くことを示している．この選好は，単語埋め込みから階層コードへの変換器を最適化する（式 3.22を最小化する）という操作と整合的である．すなわち，すべての正解の上位語に対して確率を同時に最大化する割り当てが不可能ならば，訓練データ全体に対する損失を最小化するためには，多くの下位語を抱える上位語を優先するのが合理的な解になる．あくまで直感的で厳密性を欠くたとえだが，階層コードの割り当てを，訓練

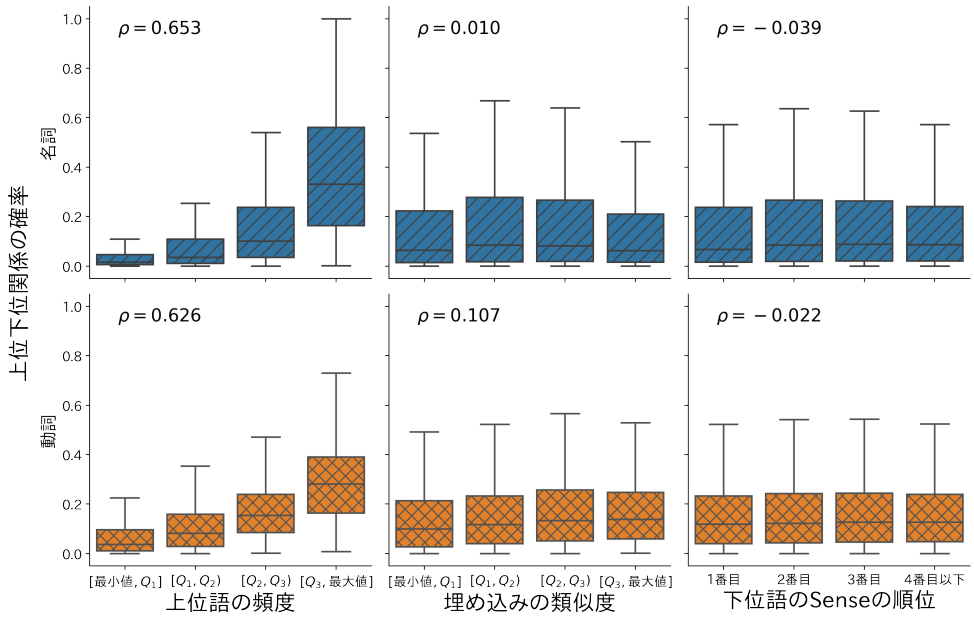


図 3.7: 上位下位関係の確率と各因子との相関．左上の数字は因子と確率のスピアマン順位相関．箱ひげ図は各因子の水準を 4 区分に分割．

データの上位下位語対から階層構造を組み立てる過程と見立てることにしよう．このとき多義語が属する枝の候補は複数ある．たとえば *mouse* ならば人物・動物・機械の枝が候補である．これらの中から，どれかひとつを残して他は捨てなくてはいけない．上述した選好は，細い・短い枝を除去して主要な太い・長い枝を保持するアプローチとして理解できる．

3.6 本章のまとめ

本章では，学習ずみの静的単語埋め込みを意味の概念階層に適応させる手法の提案および，適応によって上位下位関係識別タスクの性能が改善できることを報告した．提案手法は，単語埋め込みを階層コード表現に変換するモデルアーキテクチャおよび，階層コードのペアに対して包含関係を微分可能な形式で計量する手法からなる．これらを組み合わせて用いることにより，意味の概念階層が持つ推移律および反対称律に沿うような推論および，WordNet などの語彙資源から得られる上位下位語対を教師信号として変換モデルを学習することが可能になる．

提案手法の有効性を示すため，学習したモデルで変換した階層コードを用いて上位下位関係識別タスクを解いた．その結果，提案手法は上位下位関係の分類タスクにおいて，既存手法を上回る性能であることを報告した．タスクの誤り分析からは，全体部分関係および対義関係の誤差が大きいこと，またその理由の一部は，概念の抽象度は捉えられているが，上位下位関係と誤認するためであるとわかった．こうした特徴は，手法そのものの性質というよりも，学習に用いる語彙知識をどこまで与えるかが要因になっている可能性がある．具体的には，これらの誤認しやすい意味関係を語彙資源から取得して，上位下位語ペアでないことを明示的に学習させることで性能が改善する可能性がある．提案手法の有効性分析からは，再構築損失による補助タスクおよび最近傍語による負例の生成といった工夫が，性能向上への寄与が大きいことがわかった．これらの工夫が性能向上に寄与する理由は，全体部分関係や同位関係のように，単語埋め込みの類似度は高いが上位下位ではない意味関係の識別精度を改善するためだと示唆された．一方で，訓練データを直接の上位下位語ペアのみに制限した学習では性能が低下することが示された．この結果は，間接の上位下位関係が正確なコード表現への変換に寄与することを示している．

提案手法によって単語埋め込みが意味の概念階層に適応したかを調べるため，階層コードの割り当て方を WordNet の概念階層と比較分析した．具体的には，離散化した階層コードが単語の意味によるクラスタリングであると見立てて，WordNet の上位下位関係および同義関係によって構成される Synset の階層構造と比較した．その結果，階層コードが同一の単語は，単語埋め込みが似ているだけではなく，概念階層上でも近いことが確認できた．一方で階層コードは概念階層を正確に反映するわけではなく，改善の余地は大きいことも示された．たとえばコードが同一であることはしばしば同位関係を示唆するが，同義関係を表すとはいえない．

上位下位関係識別は単語の多義性を考慮しないタスクであるため，提案手法では静的単語埋め込みを用いている．一方で階層コード表現への変換モデルの訓練時には，多義語に対しては複数の単語と形成した上位下位語ペアが正解として与えられる．このため，多義語の埋め込みを階層コードに変換する過程では，複数の正解の中から一部を選好する操作が暗黙に行われている．階層コードから計算される上位下位関係の確率を分析した結果，多数の下位語を抱える上位語を選好する傾向が示された．

本章の提案手法は概念階層への適応および上位下位関係識別への応用に特化しているが，その帰結として，ふたつの課題を伴っている．ひとつは多義性の取り扱いである．上位下位関係識別では文脈から独立した単語固有の意味を扱うが，語義曖昧性解消など多くの応用タスクでは，文脈に依存して決まる単語の意味を扱う必要がある．もうひとつは多様な意味関係の活用である．意味の階層性は重要な性質ではあるが，概念階層を持つ品詞は名詞および動詞のみであり，形容詞や副詞では非階層的な意味関係が概念を体系化する中核である（§ 2.3.1）．これらの課題を踏まえて，次章では，文脈依存埋め込みの意味ネットワークへの適応による語義曖昧性解消に取り組む．

第 4 章

意味ネットワークへの適応による語義曖昧性解消

本章では，語義曖昧性を解消したい対象語の埋め込み，および語釈文から計算した語義の埋め込みを，単語の意味と語義の意味関係からなるネットワークの構造に適応させる手法を提案する．そして，適応させた埋め込みを用いた最近傍法によって，文脈に合致する単語の意味を選ぶ問題である語義曖昧性解消タスクの性能が改善できるかを検証する．

4.1 概要

文中の対象語について，文脈に照らして正しい意味を特定するタスクは語義曖昧性解消（WSD）と呼ばれ，評判分析 (Sumanth and Inkpen 2015, Hung and Chen 2016)，情報検索 (Zhong and Ng 2012)，機械翻訳 (Campolungo et al. 2022) など，文の意味理解が求められる多くの場面で応用されている．なかでも本研究の提案手法は，語彙資源のみを用いる知識ベース WSD というアプローチに属する．このアプローチは，語義注釈付き用例文を用いる教師あり WSD に対して精度は劣るが，単語の用例文に正解語義を注釈する手間が不要であるため，多言語対応や実用化に有利である (Bevilacqua et al. 2021)．

知識ベースアプローチの有望な方法論は，文脈依存埋め込みを用いた最近傍法である (Wang and Wang 2020)．これは，BERT（§ 2.2）を用いて用例文内の対象語および語義の語釈文をそれぞれ埋め込みに変換—対象語埋め込み，および語義埋め込みと呼ぶ—したうえで，対象語にもっとも近い語義を選択する方法

である．最近傍法の鍵は，語釈文と実際の用例文との対応付けである．BERT が計算した対象語埋め込みは，粗い粒度で語義を区別できるにとどまる（§ 2.2.3）ことから，対象語と正解語義の埋め込みを近付ける何らかの手段があれば，さらに性能が伸びると考えられる．実際に先行研究 (Wang and Wang 2021) では，意味的に関連する語義どうしを近付けると結果的に正解語義が最近傍になる確率が高まり，WSD タスクの精度が向上することを報告した．一方でこの手法は，単語の意味および語義どうしの意味関係が形成する意味ネットワークの観点からみると，ネットワーク上で互いに隣接する語義の情報のみを使っており，隣接しない語義や，単語とそれに隣接する語義の情報は活用されていない．

本章では，単語および語義からなる意味ネットワークの構造に適応するように，対象語および語義の埋め込み間の近さ・遠さを変更する手法を提案する．また適応ずみの埋め込みを用いた最近傍法によって，WSD タスクの精度が改善するかを実験する．提案手法の模式図を図 4.1に示す．

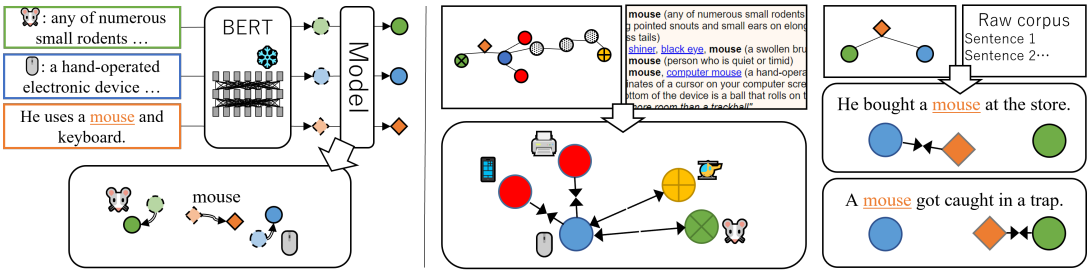


図 4.1: 提案手法の概要．語釈文と対象語の BERT 埋め込みを変換して，埋め込み間の近さ・遠さを変更する（左）．推論時は，変換した（適応ずみの）埋め込みを用いて対象語の最近傍語義を選択する．変換関数の最適化は，吸引・反発学習（中）と自己学習（右）を用いる．図中の◇は単語，○は語義を表す．

提案手法の要点は，BERT が計算した埋め込みを変換する関数を，吸引・反発学習および自己学習の併用によって最適化することである．まず吸引・反発学習では，意味ネットワーク上での語義どうしの結びつきに関する特徴を教師信号として，語義間の類似度を変更する．具体的には，隣接する語義（例：“入力機器のマウス”—“プリンタ”）を近付ける．同時に，単語を介して 2 ホップでつながる語義（例：“入力機器のマウス”—*mouse*—“動物のマウス”）および，隣接と 2 ホップのどちらにも該当しない語義（例：“入力機器のマウス”, “ヘリコプター”）をそれぞれ遠ざける．これらの操作は，意味関係でつながる，つまり意味的に関連する語義を近付けつつ，同じ単語の異なる語義や意味的に関連しない語義を

遠ざけることになる．つぎに自己学習では，平文コーパスを訓練データとして，対象語と語義の類似度を変更する．課題は，どの語義と近付けるかである．多義語は複数の語義と隣接する，つまり 1 ホップでつながる語義が複数あるが，対象語の文脈に合致する語義のみと近付けるべきである．しかし平文コーパスには正解語義が注釈されていない．そこで自己学習では，埋め込みの類似度によって対象語にもっとも近い語義と近付ける．これは最近傍法による予測語義を擬似正解とみなしていることに相当するので，ブートストラッピングだといえる．吸引・反発学習と自己学習を併用することで，語義対および単語・語義対の両方について類似度を変更し，意味ネットワークの構造に適応させる．これにより，対象語に対して正解語義を候補語義のなかで最近傍に配置する狙いである．

提案手法の有効性は，適応ずみの埋め込みを用いた最近傍法による WSD タスクの性能を実験することで検証する．また有効性の要因分析によって，どのような学習が性能に影響しているのかを調べる．さらに意味ネットワーク構造への適応を検証するために，適応ずみ埋め込みでは意味的に関連する・しない単語と語義や語義間の類似度がどのように変化したかを分析する．また，類似度の変化と WSD タスク性能の関連性を観察し，埋め込みの類似度を意味ネットワーク構造の特徴に合わせることが WSD に寄与するのかを考察する．

4.2 関連研究

語義曖昧性解消（WSD）の研究は，SemCor（§ 2.3.2）のような語義注釈付き用例文コーパスを用いる教師あり WSD と，WordNet のような語彙資源のみを用いる知識ベース WSD のふたつに大別される．本研究は後者に属している．

4.2.1 知識ベース語義曖昧性解消

知識ベース WSD では，古くから語釈文と対象語の文脈との意味的またはトピック的な類似性が使用されてきた (Lesk 1986)．たとえば *bank* は “土手” の語義ならば水や川，“銀行” の語義ならば預金や金融といったキーワードが語釈文に含まれることから，文脈に現れるトピックは語義を選択する手がかりになる．これに対して文脈依存埋め込みは周辺の単語を考慮して対象語の埋め込みを計算する仕組みであり，実際に BERT 埋め込みは同じ単語でも文脈による意味や用途

の違いを粗い粒度で捉えているとの報告がある (Reif et al. 2019, Loureiro et al. 2021).

かかる背景のもと， Wang and Wang (2020) は SREF という手法を提案し，BERT 埋め込みによる最近傍法が有望だと報告した．SREF は，BERT を用いて，レンマ・語釈文・用例の連結文をあらかじめ語義埋め込みに変換しておき，対象語の埋め込みに最も近い語義を選択する方法論である．また著者らは，埋め込み空間上で対象語を正解語義に近づけるには，上位下位関係など意味的に関連する語義どうしを加重平均することで，語義埋め込みを意味適応させることが有効だと報告した．

これを発展させた COE (Wang et al. 2021) は，同一文書内で単語の意味は一貫するという仮説 (Gale et al. 1992) に立脚して，文書レベルの文脈情報を用いて対象語埋め込みを強化する方法である．具体的には，評価対象文を前後の文とまとめて BERT に入力する，文書内で共起する語義の埋め込みを連結するなどの工夫を行う．しかしこの方法は，SNS 上の短文やレビュー，検索クエリなど，文単体で処理する必要がある用途には適用できず，汎用性に欠ける．実際に，語義曖昧性解消タスクにおいて文書情報が利用できない評価データセットは珍しくない (Pasini et al. 2021, Campolungo et al. 2022)．これに対して本研究では，文書情報を用いるのではなく埋め込みの意味適応によって性能を改善する方法を提案する．

知識ベース WSD におけるもう一つの研究の方向性は，最近傍法による語義選択を改善する経験則の開発である．Try-again Mechanism (TaM: Wang and Wang (2020), Lacerra et al. (2020)) は，候補語義の Supersense（§ 2.3.1）に属するすべての語義，つまり抽象化すれば同義とみなせる語義との類似度を考慮して候補語義を選び直すと，精度が向上するという経験則である．また，Supersense のかわりに Coarse Sense Inventory (CSI，§ 2.3.2) を用いることで計算手順を簡素化した派生版も提案されている (Wang and Wang 2021)．TaM は対象語と語義の類似度を所与として候補語義を選び直す経験則であり，埋め込みの計算方法とは無関係であることから，提案手法でも併用可能である．そこで本研究では，提案手法においても TaM との併用が有効か，および性能の改善幅を評価する．

4.2.2 教師あり語義曖昧性解消

教師ありアプローチでは，SemCor のような語義注釈付き用例文コーパスに依存して語義曖昧性解消を解く．一方で，語義注釈は人手のかかる作業であるためコーパスに採録される単語や語義は限られること，また使用頻度の高い語義に偏るという課題が指摘されている (Pasini 2020)．このような課題への対策として，教師あり WSD においても語彙資源を利用する手法が研究されてきた．たとえば Barba et al. (2021b,a) は，用例に語釈文を連結したテキストから，正解語義の語釈文スパンを抽出するという新しい解き方を提案した．WSD をスパン抽出タスクとして解くことは，用例と語釈文を連結して入力することでモデルに相互作用を考慮させられるという長所があり，有望な方法論とみなされている．また埋め込みの類似度に基づく方法論は，教師あり WSD でも用いられている．`Sup-kNN` (Supervised k-nearest neighbors) (Loureiro and Jorge 2019) は，対象語埋め込みを注釈語義（正解語義）ごとに平均化したものを語義埋め込みとみなして，最近傍法により問題を解く手法である．`BEM` (Bi-Encoder Model) (Blevins and Zettlemoyer 2020) は，対象語が正解語義に近づくように，語釈文および対象語を埋め込みに変換する BERT をファインチューニングする手法である．本研究の提案手法は，対象語と正解語義を近付けようとする点では `BEM` と共通するが，BERT をファインチューニングするのではなく BERT が計算した埋め込みを適応させるという点で異なっている．逆に言えば，提案手法の性能を `Sup-kNN` および `BEM` と比較することは，語義注釈付き用例文コーパスを使用しないこと，および BERT をファインチューイングしないことが性能に与える影響を考察する手がかりになる．

提案手法で用いる自己学習の基礎となるブートストラッピングは，教師あり WSD でも研究されてきた方法論である．たとえば，語義注釈のコストを削減することを目的として，分類器が予測した語義を正解とみなして，その分類器自体を再学習する手法が提案されている (Navigli 2009)．

4.2.3 埋め込みの意味適応

計算ずみ埋め込みを語彙知識に適応させるように更新または変換する方法論は，主に静的単語埋め込みに意味関係的知識を反映する手段として研究されてきた（§ 2.4.1）．たとえばVulic and Mrksic (2018), Mrksic et al. (2017) は，上位下位

語および同義語を近付けて，対義語を遠ざける手法を提案し，単語間の意味関係を識別するタスクの性能が向上することを報告した．これは提案手法が用いる吸引・反発学習とおなじコンセプトの操作だといえる．既存研究に対する提案手法の主な相違点は，静的単語埋め込みではなく，語義埋め込みおよび，文脈によって変化する対象語埋め込みを扱うことである．また対象語に対して近付けるべき語義（正解語義）を指示する必要性が生じることが，自ら用意した擬似正解を使うという自己学習を導入する動機となっている．

画像やテキストを表すベクトルを獲得する表現学習の研究では，対照学習の有効性が多く報告されている (Chen et al. 2020, Gao et al. 2021, Wang et al. 2021, Giorgi et al. 2021)．対照学習は，対象の画像やテキストを正例と近付けつつ負例と遠ざける手法であり，吸引・反発学習で行いたい操作と合致する．そこで提案手法では，この対照学習の概念を利用して，吸引・反発学習の目的関数を対照損失を用いて定式化する．なお対照学習においては，負例を，強負例（hard negatives）とバッチ内負例（in-batch negatives）の 2 種類に分けることがある．前者は正例と見分けにくい事例であり，一般に教師信号を用いて作成される．後者は正例と見分けやすい事例であり，文字通り乱択した事例の集合から自動的に作成される．これらは提案手法の吸引・反発学習にもあてはめることが可能で，おなじ単語の異なる語義（異義）は強負例とみなせるし，意味的関連性がない語義（無関連語義）はバッチ内負例と同義である（§ 4.3.4）．

4.3 提案手法

提案手法は，BERT を用いてあらかじめ計算した語義埋め込みおよび対象語埋め込みを変換し，意味ネットワークの構造に適応させるものである．

$$\boldsymbol{v}_w = H_w(\hat{\boldsymbol{v}}_w) \tag{4.1}$$

$$\boldsymbol{e}_s = H_s(\hat{\boldsymbol{e}}_s) \tag{4.2}$$

関数への入力 $\hat{\boldsymbol{v}}_w$ および $\hat{\boldsymbol{e}}_s$ は，語義曖昧性解消の対象語 w および語義 s の BERT による埋め込み，出力 $\boldsymbol{v}_w$ および $\boldsymbol{e}_s$ は適応ずみの埋め込み，H_w および H_s は変換関数である．提案手法は訓練と推論のふたつのフェーズからなる．訓練フェーズでは，変換関数を最適化することを目的とし，$\boldsymbol{v}_w$ および $\boldsymbol{e}_s$ によって定義され

る吸引・反発学習および自己学習の損失の加重和を最小化する．BERT 自体はファインチューニングされず，提案手法は BERT の埋め込みを変換する仕組みである．また埋め込みの更新ではなく変換関数を学習することによって，推論フェーズで任意の対象語埋め込みを適応させることができる．語義埋め込みの計算や変換関数の訓練に用いる語彙資源は，WordNet から取得する．

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．語義の識別子は WordNet の Sense（§ 2.3.1）である．具体的には，曖昧性を解消する対象語 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．

$$s^* = \arg\max_{s' \in \mathcal{S}_w} \rho_{w,s'} \tag{4.3}$$

$$\rho_{w,s} = \cos(\boldsymbol{v}_w, \boldsymbol{e}_s) = \frac{\boldsymbol{v}_w \cdot \boldsymbol{e}_s}{\|\boldsymbol{v}_w\| \|\boldsymbol{e}_s\|} \tag{4.4}$$

なお最近傍法で語義曖昧性解消を行う際には，Try-again Mechanism (TaM) という経験則を併用することで性能が向上するとの報告がある (Wang and Wang 2021, 2020, Wang et al. 2021)．TaM は対象語と語義の類似度を活用して候補語義を選び直す経験則であり，埋め込みの計算方法には依存しない．そこで本研究では，提案手法と TaM を併用する場合としない場合の両方について結果を報告する．また TaM 経験則の概要は § 4.3.5で説明する．

以下の項では，まず BERT を用いて語義埋め込みおよび対象語埋め込みを計算する手順および，計算に用いる語彙資源を説明する．次に本研究の提案である，吸引・反発学習および自己学習を用いて変換関数を最適化する手法を定式化する．また，提案手法の名称を `SS-WSD`: Semantic Specialization for WSD とする．

4.3.1 BERT による埋め込みの計算

BERT を用いて文脈依存単語埋め込みを計算する手順は，先行研究 (Wang et al. 2020, Bevilacqua and Navigli 2020, Wang and Wang 2020) に倣う．具体的には，埋め込みを計算したい単語 w_t を含む単語列の先頭および末尾に特殊トークンである `[CLS]` および `[SEP]` を付与したうえで BERT に入力し，Transformer アー

キテクチャの最上位 4 層における出力の和を取得する．

$$\hat{\boldsymbol{v}}_{w_t} = \sum_{l=L-3}^{L} \boldsymbol{z}_t^l \tag{4.5}$$

ここで $\boldsymbol{z}_t^l$ は，単語 w_t に対応するトークンの，全部で L 個ある Transformer アーキテクチャ中間層のうち l 層目の出力である（§ 2.2.2）．なお単語が部分単語（サブワード）に分割されている場合は，部分単語の平均を $\boldsymbol{z}_t^l$ として用いる．

BERT モデルは，`transformers` パッケージが提供する `bert-large-cased` を用いる．埋め込みの次元数は 1,024，Transformer 層数は 24，入力テキストは大文字・小文字を区別する．

4.3.2 語義埋め込みの計算

提案手法が使用する語彙資源は WordNet であり，Sense を語義の識別子とみなす．すなわち提案手法による語義曖昧性解消タスクは，対象語の Lemma で WordNet を検索したときに返される Sense の配列からひとつを選ぶ問題として定義される．また語義埋め込みは Sense ごとに与えられる．

語義埋め込みの計算手順は，先行研究 (Wang and Wang 2020) に従う．具体的には，Sense に紐付く Lemma および，Sense が属する Synset から取得した Gloss（語釈文）・同義語の Lemma（レンマ）n 個・Example（例文）m 個を連結した文を作成し，これを BERT で単語埋め込みにエンコードする（§ 4.3.1）．そして特殊トークンを含むすべての単語 $w_0 w_1 ... w_T w_{T+1}$ に対する平均を，語義埋め込みとする．

$$\hat{\boldsymbol{e}}_s = \frac{1}{T+2} \sum_{t=0}^{T+1} \hat{\boldsymbol{v}}_{w_t} \tag{4.6}$$

連結文のテンプレートを以下に示す．テンプレート内の `n` および `m` は同義語および例文の数である．

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

例として，単語 *computer* の語義 “計算機” にあたる `computer%1:06:00::` をテンプレートにあてはめた場合の連結文を以下に示す．また，元になった WordNet の要素を表 4.1に示す．

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

表 4.1: 語義 `computer%1:06:00::` に対する WordNet の要素

要素	値
Sense	`computer%1:06:00::`
Lemma	*computer*
Synset	`computer.n.01`
Gloss	*a machine for performing calculations automatically*
Example	`Not Available`
同義語	*computer, computing device, data processor, ...*

なお例示した語義の Synset には Example（例文）がない．実際に Synset の大半は例文がないか，あっても短文である（§ 2.3.1）．先行研究 (Wang and Wang 2020) はこれを問題視して独自に収集した例文を併用しているが，本研究では WordNet に収録された例文のみを使用する．

4.3.3 変換関数

埋め込みを適応させる変換関数は，語義向けの H_s と対象語向けの H_w のふたつを定義する（式 4.2）．アーキテクチャはいずれも同一であり，残差接続を採用する．また残差接続の関数は，ReLU を活性化関数とする 2 層の順伝播型ニューラルネット FFNN を用いる．

$$\boldsymbol{v}_w = H_w(\hat{\boldsymbol{v}}_w) = \hat{\boldsymbol{v}}_w + \epsilon\|\hat{\boldsymbol{v}}_w\|F_w(\hat{\boldsymbol{v}}_w) \tag{4.7}$$

$$\boldsymbol{e}_s = H_s(\hat{\boldsymbol{e}}_s) = \hat{\boldsymbol{e}}_s + \epsilon\|\hat{\boldsymbol{e}}_s\|F_s(\hat{\boldsymbol{e}}_s) \tag{4.8}$$

$$F_w(\cdot) = 2\sigma(\mathrm{FFNN}_w(\cdot)) - 1 \tag{4.9}$$

$$F_s(\cdot) = 2\sigma(\mathrm{FFNN}_s(\cdot)) - 1 \tag{4.10}$$

σ はシグモイド関数であり，残差接続 $F_{\{w,s\}}$ の各次元要素の出力を $[-1, 1]$ に収める．ϵ は変換による移動すなわち変換前後の距離を制約するハイパーパラメータである．具体的には，埋め込みの次元数を N_d とすると，変換前の BERT 埋め込みからの相対 L2 距離を $\|\boldsymbol{v}_w - \hat{\boldsymbol{v}}_w\|/\|\hat{\boldsymbol{v}}_w\| \leq \epsilon\sqrt{N_d}$ に抑える役割を担う．

距離制約の導入は自己学習（§ 4.3.4）の促進を意図している．後述のとおり，自己学習は変換関数の最適化が進むにつれて対象語の最近傍に正解語義がくる

精度が改善するというブートストラップ的な目的関数であるが，不正解が最近傍にくると逆向きに作用する恐れもある．このため，距離制約によって変換前の BERT 埋め込みの類似度特性を適度に維持することで，不正確な語義を擬似正例に選ぶことによる精度低下の悪循環を回避する．距離制約を正当化する観察結果として，最近傍語義を正解と入れ替えるために最低限必要な埋め込み類似度の変化は小さいことを報告する（§ 4.6.1）．また，距離制約の有効性を実験的に検証した結果を報告する（§ 4.6.3）．

4.3.4 訓練の目的関数

変換関数の最適化に用いる目的関数は，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．

$$L = L^{\mathrm{AR}} + \alpha L^{\mathrm{ST}} \tag{4.11}$$

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．ふたつの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類似度を調節する働きをするが，語義と単語の類似度についての教師信号はもたらさない．一方で自己学習は，文脈をふまえて対象語の意味にもっとも近い語義と近付けるが，それ単体ではもともと BERT が近いと判断している語義をさらに近付けるだけなので，最近傍語義を正解と入れ替える働きは限られるはずである．ふたつの学習を併用する効果は，実験的に検証する（§ 4.6.2）．

吸引・反発学習

吸引・反発学習は，WordNet にある意味関係を教師信号として，意味的に関連する語義対を近付けつつ，関連しない語義対および，同じ単語に接続するが意味が異なる語義対を遠ざける．学習に用いる語義対は，WordNet の意味関係（§ 2.3.1）を用いて定義する．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 種類の語義集合を定義する．

- 関連語義 $\mathcal{S}_s^P$ は，上位下位や全体部分などの意味関係で結ばれる語義である．意味ネットワーク上では，隣接する語義（例：“計算機”—“機械”）にあたる．

正確な定義は，先行研究 (Wang and Wang 2020) の論文および実装[1] に従う．具体的には，まず Derivationally Related Form（§ 2.2）を用いて，語形変化の関係にある Sense を取得して集合を作る．次に，それぞれの Sense が属する Synset を収集して集合に追加する．さらに，表 4.2に示す意味関係でつながる Sense および Synset を収集して集合に追加する．最後に，Synset に属する Sense を取得する．

- 異義 $\mathcal{S}_s^N$ は，同じ Lemma に結びつく異なる語義，すなわち多義語の語義から自分自身である s を除いたものである．意味ネットワーク上では，単語を介して 2 ホップでつながる語義（例：“計算機”—*computer*—“計算手”[2]）にあたる．

- 無関連語義 $\mathcal{S}_s^U$ は，すべての Sense（意味目録）から乱択した語義である．意味ネットワーク上では，ほぼ確実に隣接でも異義でもない語義（例：“計算機” と “動物のマウス”）にあたる．

表 4.2: 関連語義の収集に用いる WordNet の意味関係

要素	意味関係
Sense	Pertainym, Antonym, Synonym
Synset	Hypernym, Hyponym, {Holonym, Meronym}—{Part, Member, Substance}, Domain—Usage, Entails, Causes, See also, Attribute, Similar to

具体例として，“計算機” の語義にあたる `computer%1:06:00::` に対する関連・異義・無関連語義の一部を表 4.3に示す．

吸引・反発損失 L^{AR} の定式化は，対照損失（Contrastive loss: Gao et al. (2021), Wang et al. (2021), Giorgi et al. (2021)）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，勾配降下法で用いる乱択ミニバッチを $\mathcal{S}^B$ として，そのうちのひとつの語義 $s \in \mathcal{S}^B$ 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．損失

[1] https://github.com/lwmlyy/SREF
[2] 単語 *computer* における “計算手” の語義を表す．

表 4.3: 語義 computer%1:06:00:: に対する関連・異義・無関連語義の例

種類	WordNet 上の関係	語義（Sense）
関連語義 $\mathcal{S}_s^P$	Synonym	computing_device%1:06:00::
	Hyponym	analog_computer%1:06:00::
	Hypernym	machine%1:06:00::
	Meronym—Part	monitor%1:06:02::
異義 $\mathcal{S}_s^N$	Lemma が一致	computer%1:18:00::[2]
無関連語義 $\mathcal{S}_s^U$	なし	goldfish%1:05:00::
	なし	chef%1:18:01::

関数 L^{AR} の定義は以下のとおりである．

$$L^{\mathrm{AR}} = -\sum_{s\in\mathcal{S}^B} \ln \frac{\exp(\beta\rho_{s,s_p})}{\sum\limits_{s'\in\left(\{s_p\}\cup\mathcal{S}_s^U\cup\tilde{\mathcal{S}}_s^N\right)} \exp(\beta\rho_{s,s'})} \tag{4.12}$$

$$\rho_{s,s'} = \cos(\mathbf{e}_s, \mathbf{e}_{s'}) \tag{4.13}$$

β は温度（temperature）と呼ばれるハイパーパラメータである．距離学習の先行研究 (Deng et al. 2019, Wang et al. 2018) にならい，$\beta = 64$ に設定する．

自己学習

自己学習は，平文コーパスを訓練データとして利用する．具体的には，コーパスから取り出した用例文に含まれる対象語について，意味ネットワーク上で隣接する語義と近付ける．しかし対象語が多義語の場合は複数の語義と隣接するので，対象語の埋め込みに最も近い語義を擬似正解とみなして近付ける．推論フェーズでは最近傍法によって予測語義を決めることを思い出そう．これとおなじく自己学習は最近傍語義を正解とみなすので，自己学習は自身の予測を正解とするブートストラップ法といえる．対象語に隣接する語義は，WordNet を用いて調べる．具体的には，対象語 w の Lemma（正規化された表層形および品詞）に紐付くすべての Sense を候補語義の集合 $\mathcal{S}_w$ として取得する．

自己学習損失 L^{ST} の定式化は，候補語義 $\mathcal{S}_w$ のうち，対象語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みなすように定義する．具体的には，対象語 $w \in \mathcal{W}^B$ の候補語義 $\mathcal{S}_w$ それぞれに対して対象語

埋め込みとの cosine 類似度を計算し，候補内での最大値を取る．

$$L^{\mathrm{ST}} = \sum_{w \in \mathcal{W}^B} (1 - \max_{s \in \mathcal{S}_w} \rho_{w,s}) \tag{4.14}$$

$$\rho_{w,s} = \cos(\boldsymbol{v}_w, \boldsymbol{e}_s) \tag{4.15}$$

最近傍語義すなわち擬似正解は，変換関数 H_w および H_s が出力する適応ずみ埋め込み $\boldsymbol{v}_w$ および $\boldsymbol{e}_s$ に依存するため，訓練が進むにつれて変化する．擬似正解を固定しない狙いは，前述したとおりブートストラップの効果を活用するためである．すなわち，訓練が進行して語義曖昧性解消の精度が向上する過程で，擬似正解の精度も同時に向上するという好循環を期待している．

4.3.5 Try-again Mechanism (TaM) 経験則

Try-again Mechanism (TaM) は，単語と語義の類似度に基づく最近傍法と併用することで，精度が向上すると報告されている経験則である (Wang and Wang 2021, 2020, Wang et al. 2021)．具体的には，類似度によって候補語義を 2 つに絞ったあとで，各語義を抽象化した概念との類似度を考慮して再順位付けする処理である．TaM はいくつかの派生版があるが，本研究では簡便性に優れる Wang and Wang (2021) のアルゴリズム[3] を使用する．

TaM 経験則のアルゴリズムは，まず対象語 w に対する類似度 $\rho_{w,s}$（式 4.4）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，§ 2.3.2) (Lacerra et al. 2020) の意味カテゴリとの類似度を計算する．これをもとの類似度に加算して更新された類似度 $\rho^+_{w,s}$ を計算し，最大の $\rho^+_{w,s}$ を持つ語義を選択する．

$$\rho^+_{w,s} = \rho_{w,s} + \max_{s' \in \mathcal{S}^{\mathrm{CSI}}_s} \rho_{w,s'} \tag{4.16}$$

$\mathcal{S}^{\mathrm{CSI}}_s$ は，語義 s と同じ CSI の意味カテゴリに属する語義の集合である．ただし s が CSI に未採録の場合は空集合，複数の意味カテゴリに属する場合はそれらの和集合とする．CSI は WordNet の Synset を大きなカテゴリで抽象化した意味目録であるため，式 4.16の右辺第 2 項は，抽象化すれば同じだといえる語義との

[3] https://github.com/lwmlyy/SACE

類似度だと解釈できる．

4.4 実験設定

4.4.1 訓練

吸引・反発学習の訓練データとなる WordNet から収集した語義対について，統計量を示す．まず，語義の異なり数は 206,949 である．またひとつの語義に対する関連語義数および異義数の平均値を表 4.4に示す．全品詞合計について数値を見ていくと，関連語義数は 8.1 であり，Synset が持つ意味関係の平均値である 2.5（表 2.3）の 3 倍以上である．この主な要因は，語形変化の関係である Derivationally Related Form を経由して，異なる品詞からも意味的に関連する語義を収集するためである（§ 4.3.2）．なお異義数の平均値は 1.3 であり，関連語義と比較すると小規模であることがわかる．

表 4.4: 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
Lemma 数	117,798	11,529	21,479	4,481	155,287
Sense 数	146,320	25,047	30,002	5,580	206,949
関連語義 $\mathcal{S}_s^P$	7.8	13.0	6.2	3.9	8.1
異義 $\mathcal{S}_s^P$	0.8	4.1	1.2	0.7	1.3

自己学習の訓練データとなる平文コーパスは，語義注釈付き用例文コーパスである SemCor（§ 2.3.2） (Miller et al. 1993) を用いた．ただし自己学習に正解語義は不要なので，注釈ずみの語義は使用しない．WordNet から候補語義を検索するための Lemma は，対象語に注釈ずみのものを使用する．対象語の数は名詞，動詞，形容詞，副詞がそれぞれ 87,002，88,334，31,753，18,947 個であり，全品詞合計で 226,036 個である．自己学習の訓練データは対象語の見出し語および品詞がわかればよいので，形態素解析・複単語表現解析ずみの任意の平文コーパスを使用できる．しかし本実験では既存研究との公平な比較を重視して，語義曖昧性解消タスクで広く普及している SemCor を用いることとした．

変換関数の最適化について述べる．アルゴリズムは Adam (学習率 0.001) ，エポック数は 15 とした．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調整の方法について述べる．ハイパーパラメータの決定は，開発データによるハイパーパラメータ最適化によって実施した．具体的には，まず，optuna パッケージ (Akiba et al. 2019) に実装されている TPESampler アルゴリズムを用いて，パラメータ空間 $N_B \in \{64, 128, 256, 512, 1024\}$，$\alpha \in [0.1, 10]$，$\epsilon \in [0.001, 0.1]$ を探索した．つぎに $\epsilon \in [0.01, 0.02]$ を刻み幅 0.001 でグリッドサーチした．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．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 (Pasini et al. 2021) の慣習にならい，Unified Evaluation Framework for WSD (§ 2.3.2) のサブセットである SemEval-2007 (SE07，Pradhan et al. (2007)) を使用した．

4.4.2 評価

語義候補の中からひとつの語義を選択する推論アルゴリズムは，TaM 経験則が無効の場合は式 4.4，有効の場合は式 4.16である．語義曖昧性解消タスクの評価データおよび方法は，標準評価プロトコル (Raganato et al. 2017) に従った．評価データセットは，Unified Evaluation Framework for WSD である．評価指標はマイクロ F 値である．なお提案手法は複数の語義を予測することはないので，マイクロ F 値は適合率および再現率と一致する (Pasini et al. 2021)．変換関数の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する．

4.4.3 ベースライン

提案手法と比較するベースラインは，先行研究を 3 つの区分にわけて整理する．ひとつめの区分は，知識ベースアプローチかつ，TaM 経験則（§ 4.3.5）を併用しない手法である．具体的には，$\texttt{WN1}^{\texttt{st}}$ (Raganato et al. 2017)，BERT ，および $\texttt{SREF}_{\texttt{emb}}$ (Wang and Wang 2020) と比較する．$\texttt{WN1}^{\texttt{st}}$ は，WordNet を Lemma で検索したときに返される Sense 配列の先頭を選ぶ．Sense はおおむね出現頻度順に整列されている（§ 2.3.1）ことから，使用頻度がもっとも多い語義を選ぶ経験則だといえる．$\texttt{WN1}^{\texttt{st}}$ は非常に単純な手法だが，知識ベース WSD では強力なベースラインとして長く用いられてきた (Navigli 2009)．BERT および $\texttt{SREF}_{\texttt{emb}}$ は，いずれも埋め込みによる最近傍法である．BERT は，BERT が計算した対象語および語義の埋め込み，すなわち適応させる前の $\hat{\boldsymbol{v}}_w$ および $\hat{\boldsymbol{e}}_s$ をそのまま用いる．

$\texttt{SREF}_{\texttt{emb}}$ は意味的に関連する語義どうしの埋め込みを加重平均したものである．

ふたつめの区分は，知識ベースアプローチかつ，TaM 経験則を併用する手法である．具体的には，$\texttt{SREF}_{\texttt{kb}}$ (Wang and Wang 2020) および COE (Wang et al. 2021) と比較する．$\texttt{SREF}_{\texttt{kb}}$ は，前述した $\texttt{SREF}_{\texttt{emb}}$ に TaM 経験則を併用する手法である．COE は，語義埋め込みは $\texttt{SREF}_{\texttt{emb}}$ と同一だが，対象語埋め込みは評価データセットから得られる文書レベルの文脈情報を用いて強化し，さらに TaM 経験則を併用する手法である．なお COE は，実験実施時点で知識ベース WSD の最高精度を報告している．

みっつめの区分は，語義注釈付き用例文コーパス（SemCor）を用いる教師ありアプローチで，なおかつ埋め込みによる最近傍法にもとづく手法である．具体的には，Sup-kNN (Loureiro and Jorge 2019) および BEM (Blevins and Zettlemoyer 2020) と比較する．Sup-kNN は，コーパスを用いて対象語埋め込みを計算し，これを正解語義ごとに平均して語義埋め込みを得る．語釈文などの語彙知識は使用しない．なお SemCor に出現しない対象語については $\texttt{WN1}^{\texttt{st}}$ に従って予測する．BEM は，対象語と正解語義が近づくように，用例文および語釈文をエンコードする BERT をファインチューニングする．これらの手法は SemCor 依存であり提案手法とは異なる問題設定のため，性能の比較は直接的なものではない．しかし埋め込みによる最近傍法という点は共通しているため，語義注釈付き用例文を用いないことや，BERT をファインチューニングしないことの影響について論じるうえで有用である．また BEM は教師あり手法であることから，その性能は BERT 埋め込みを用いた最近傍法にもとづく知識ベース WSD の性能を改善していくうえでの目標値としても有用である．

4.5 実験結果

表 4.5に，ベースラインおよび提案手法の WSD タスク性能を示す[4]．提案手法 $\texttt{SS-WSD}_{\texttt{emb,kb}}$ は，全事例合計（All 列）において知識ベースの先行研究（区分 KB および KB+TaM に属する手法）を上回った．特に TaM 経験則併用時の $\texttt{SS-WSD}_{\texttt{kb}}$ 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

表 4.5: 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．区分の Sup は教師あり手法，KB は知識ベース手法．提案手法の $\texttt{SS-WSD}_{\texttt{emb,kb}}$ 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, $\texttt{WN1}^{\texttt{st}}$, $\texttt{SREF}_{\texttt{kb}}$, 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	
Sup	Sup-kNN	×	76.3	73.2	66.2	71.7	74.1	—	—	—	—	73.5
	BEM	×	79.4	77.4	74.5	79.7	81.3	81.4	68.5	83.0	87.9	79.0
KB	$\texttt{WN1}^{\texttt{st}}$	×	66.8	66.2	55.2	63.0	67.8	67.6	50.3	74.3	**80.9**	65.2
	BERT	×	67.8	62.7	54.5	64.5	72.3	67.8	52.3	74.0	77.7	65.6
	$\texttt{SREF}_{\texttt{emb}}$	×	70.3	68.0	60.4	74.2	77.4	76.3	53.5	75.2	76.3	71.0
	$\texttt{SS-WSD}_{\texttt{emb}}$	×	**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)
KB + TaM	$\texttt{SREF}_{\texttt{kb}}$	×	72.7	71.5	61.5	76.4	79.5	78.5	56.6	79.0	76.9	73.5
	COE	✓	76.0	74.2	**69.2**	**78.2**	80.9	**80.6**	61.4	80.5	81.8	76.3
	$\texttt{SS-WSD}_{\texttt{kb}}$	×	**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)

イント（$\texttt{SS-WSD}_{\texttt{kb}}-\texttt{SS-WSD}_{\texttt{emb}}$）だった．

次に，TaM 経験則を使用しない場合（区分 KB）に注目する．提案手法によって BERT が計算した埋め込みを適応させる効果は +9.3 ポイント（$\texttt{SS-WSD}_{\texttt{emb}}-\texttt{BERT}$）だった．関連語義どうしを近付ける操作のみを行う先行研究 SREF の適応効果は +5.4 ポイント（$\texttt{SREF}_{\texttt{emb}}-\texttt{BERT}$）であることから，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．また提案手法による適応の効果を品詞別に調べると，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 個と全品詞中で最多であること（表 4.4），すなわち吸引・反発学習の教師信号が豊富であるという事実と整合する．

最後に，教師あり WSD の先行研究（区分 Sup に属する手法）と比較する．用例文に注釈された正解語義を用いて語義埋め込みを計算する Sup-kNN に対して，提案手法は 1.4 ポイント上回った（$\texttt{SS-WSD}_{\texttt{emb}}-\texttt{Sup-kNN}$）．このことは，適応ずみの埋め込みが対象語と同じ意味の語釈文を最近傍に配置する精度が，BERT 埋め込みがおなじ意味の対象語どうしを互いの最近傍に配置する精度よりも高い

[4] Wilcoxon の符号順位検定（両側検定）も実施した．その結果，スチューデントの両側 t 検定において p 値が 0.05 を下回るすべての事例で，Wilcoxon 検定の p 値は 0.0625（サンプル数 5 における理論的な最小値）だった．したがって，両検定の結果はほぼ一致していた．

ことを示唆している．これは，提案手法が用例文と語釈文という表現形式の違いをまたいで，埋め込み空間上で意味的な類似性を効果的に表現できているものと評価できる．対象語と正解語義を近付けるように BERT をファインチューニングする BEM に対して，提案手法は 4.1 ポイント下回った．この要因としては，提案手法が BERT をファインチューニングしない，すなわち対象語や語釈文を埋め込みにエンコードする過程は最適化しないためだと考えられる．一方で BEM では語義注釈付き用例文コーパスが必須であることを考慮すると，語彙資源のみを用いる提案手法がその性能に近づきつつあることは，実用化への貢献という観点で特筆すべき結果である．

4.6 分析

4.6.1 BERT 埋め込みの特性

提案手法は，BERT によって計算した対象語および語義の埋め込み（BERT 埋め込み）を意味ネットワークに適応させるものである．したがって，WSD タスクの性能は BERT 埋め込みの特性に影響を受ける．具体的には，自己学習における擬似正解の正確性は BERT 埋め込みによる対象語と語義の対応付け精度に依存するほか，変換関数に導入した距離制約（§ 4.3.3）は，BERT 埋め込みを大きく移動させる必要はないと暗黙に仮定している．

まず，BERT 埋め込みを用いて最近傍法で WSD タスクを解いた場合の性能を，表 4.5の手法 BERT に示す．BERT は知識ベース WSD の伝統的なベースラインである WN1^{st} を 0.4 ポイント上回ることから，BERT 埋め込みが正解語義を最近傍に配置する精度は，最頻語義を選ぶいわゆる多数決法と同等であることを示している．またこの結果は，対象語に擬似正解語義を付与する方法として，WN1^{st} よりも自己学習によるブートストラップのほうが効果的であることを支持している．

つぎに，正解語義をどれだけ近付ければ最近傍にできるのかを調べるために，BERT 埋め込みを用いて，対象語にもっとも近い不正解語義と正解語義との cosine 類似度の差（マージン）を分析する．なお分析には本実験の評価データを再利用した．マージンの累積分布を図 4.2に示す．図中の縦破線は，マージンが 0.05 の位置にある．破線と累積分布曲線の交点から，品詞を問わず約 90%

の対象語ではマージンが 0.05 以下であることがわかる．すなわち，BERT 埋め込みは既に正解語義を最近傍付近に配置できており，わずかに cosine 類似度を高めるだけで不正解語義と入れ替え可能なことを示している．この結果は，変換関数に距離制約を導入することによって，適応前の BERT 埋め込みによる対象語と語義の類似度を適度に維持することの妥当性を示している．

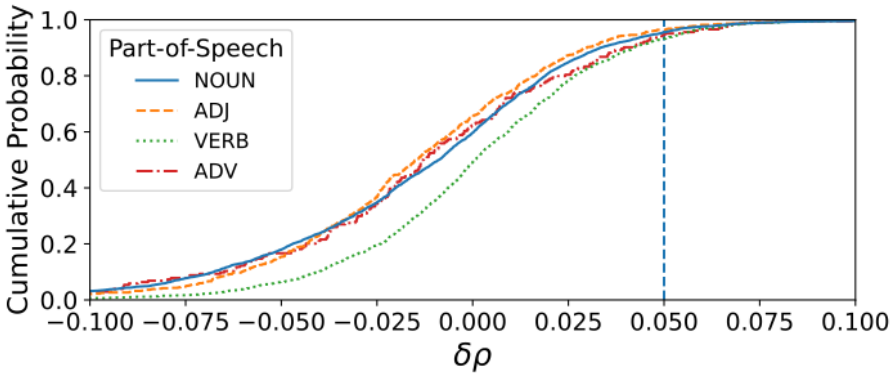


図 4.2: 不正解語義との類似度および正解語義との類似度のマージンに対する累積分布．$\delta\rho_w = \max_{s \in \mathcal{S}_w \setminus \mathcal{S}_w^{\mathrm{gt}}} \rho_{w,s} - \max_{s \in \mathcal{S}_w^{\mathrm{gt}}} \rho_{w,s}$，ただし $\mathcal{S}_w$ および $\mathcal{S}_w^{\mathrm{gt}}$ は対象語 w の候補語義および正解語義の集合．

4.6.2 目的関数の効果

提案手法の有効性を考察するために，目的関数（§ 4.3.4）などの一部を無効化にして，性能がどのように変化するかを調査するアブレーション分析を実施する．WSD タスク性能の変化を，表 4.6に示す．なお提案手法である埋め込み適応の効果に着目するために，TaM 経験則は併用しない設定で分析した．

吸引・反発学習または自己学習を無効化すると，精度はそれぞれ 3.3 および 4.4 ポイント低下する．このことから，両学習は相補的であることが示唆される．すなわち，吸引・反発学習によって意味関係的知識を語義間の類似度に反映することと，自己学習によって単語が取り得る意味を対象語と語義の類似度に反映することは，一方が他方を兼ねるのではなくそれぞれが異なる効果を発揮することがわかる．

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

表 4.6: 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$_{\texttt{emb}}$	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
– 対象語埋め込みの適応	71.7	−3.2

い，または同じ単語の異なる語義を遠ざけることはいずれも有効である．またこのことは，WordNet に直接書かれている二項関係のみを教師信号とするのではなく，意味関係によって形成されるネットワーク構造の特徴を使うことの有効性を示している．具体的には，隣接しない語義対（例：“計算機” と “動物のマウス”）や，単語を介して 2 ホップでつながる語義対（例：“計算機”—*computer*—“計算手”）を教師信号に加えて吸引・反発学習をすることが有効である．なお無関連語義の寄与が異義を上回る理由は，訓練事例数の違いが考えられる．無関連語義は乱択ミニバッチから自身を除いたものなので，$N_B - 1 = 256 - 1 = 255$ 個となる．これに対して異義は多義語の語義候補から自身を除いたものであり，平均 1.3 個（表 4.4）であり，2 桁小さい．また対照損失（式 4.12）の計算で異義の重みを大きくすることも試みたが，性能には特段影響しなかったことを付記する．

対象語埋め込みの適応をやめる，すなわち変換関数 H_w（式 4.2）を恒等写像にして $\boldsymbol{v}_w = \hat{\boldsymbol{v}}_w$ にすると，精度は 3.2 ポイント低下する．このとき目的関数は不変，つまり依然として吸引・反発学習および自己学習を併用していることに注意されたい．したがって，語義および対象語の埋め込みはいずれも適応させるべきであり，既存研究 SREF のように語義埋め込みの適応のみでは不十分だとわかる．もしも対象語と正解語義の近さだけが問題なのだとしたら，語義埋め込みの適応のみでも性能は低下しないはずだが，実際にはそうではない．このことは，対象語埋め込みの適応は，語義が異なる用例どうしを遠ざけることで識別性を改善している可能性を示唆している．なお念のために距離制約のハイパーパラメータである ϵ のチューニングをやり直してみたが，結論は変わらなかった．

4.6.3 距離制約の効果

提案手法は，変換関数に距離制約のハイパーパラメータ ϵ を導入して調整することにより，適応ずみ埋め込みがもとの BERT 埋め込みから過度に逸脱しないようにしている．距離制約の有効性を検証するため，ϵ を $[0.1, 0.2]$ の範囲で変更したときの WSD タスク精度を図 4.3に示す．また図の範囲外であるが，$\epsilon = 0.03$ まで大きくした場合は，精度は 71.5 まで低下したことを付記する．

ϵ に対して精度は逆 U 字曲線を示すことから，ϵ を小さくした厳しい距離制約と，ϵ を大きくした緩い距離制約の中間が最適であることが分かる．このことは，BERT 埋め込みによる対象語と語義の類似度を劇的に変化させるよりも，適度に維持することが有効であると示している．また距離制約が有効である理由としては，自己学習において明らかに誤った語義を擬似正解に選ぶ可能性が低下すること，および変換関数の最適化における探索空間が縮小する効果が考えられる．

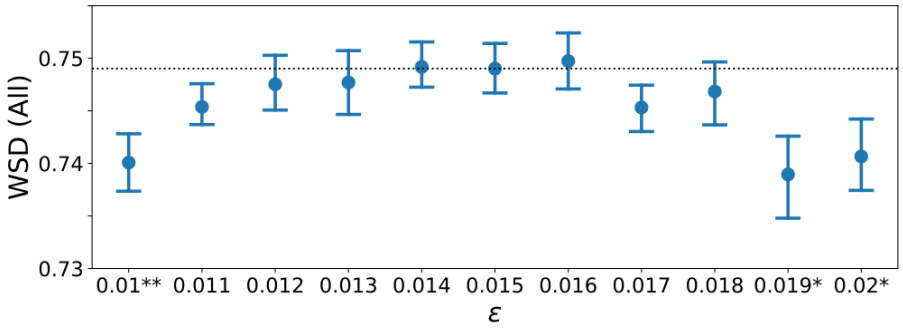


図 4.3: 距離制約のハイパーパラメータ ϵ（§ 4.3.3）と WSD タスク精度の関係．点およびエラーバーは 5 回試行の平均および標準偏差．デフォルト設定（$\epsilon = 0.015$）の精度を水平点線で表示．*はデフォルト設定時との差の統計的有意性（Welch の両側 t 検定，$*: p < 0.05$，$**: p < 0.005$）．

4.6.4 自己学習訓練データ量の影響

本実験での自己学習には SemCor コーパスを訓練データとして使用したが，原理的には (形態素解析等を施した）平文コーパスも適用可能であり，Wikipedia 等を選択することで容易に大規模化が可能である．そこで，訓練データ量の影響を評価するため，SemCor コーパスの一部のみを使用した場合の WSD タスクの精度を測定した．結果を図 4.4に示す．訓練事例数すなわち対象語の数が増えるに

つれて精度が向上し，全体の 60%にあたる 13.6 万（136k）件まで増やしたところで飽和した．したがって，自己学習に用いる事例数は 10 万件程度で十分であること，および事例数を数百万件に増やしてもスケーリングの効果は期待できないことが示された．一方で，語義曖昧性解消においては単純な規模よりも単語と語義の網羅性が問題だと指摘されている (Pasini 2020)．したがって，多様な用例を偏りなくサンプリングする工夫と併用するならば，大規模な平文コーパスの活用が有用である可能性がある．これは語義注釈付き用例文コーパスの構築という本研究とは異なる方向性になるため，今後の課題としたい．

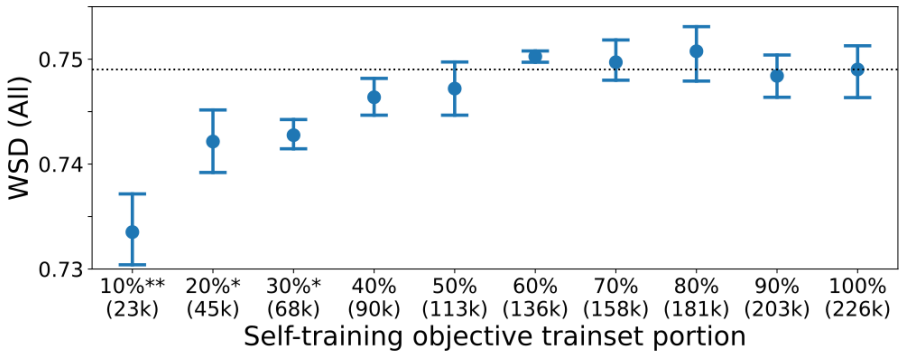


図 4.4: 自己学習の訓練データ量削減と WSD タスク精度の関係．点およびエラーバーは 5 回試行の平均および標準偏差．デフォルト設定（全データを使用）の精度を水平点線で表示．*はデフォルト設定時との差の統計的有意性（Welch の両側 t 検定，$*: p < 0.05$，$**: p < 0.005$）．

4.6.5 類似度の変化

埋め込みを適応させる前後で，関連・無関連・異義との類似度および，対象語と正解語義との類似度がどのように変化したかを，表 4.7に示す．$\rho_{\mathcal{S}^P}$, $\rho_{\mathcal{S}^U}$, $\rho_{\mathcal{S}^N}$ はそれぞれ，語義 s に対する関連語義 $\mathcal{S}_s^P$，無関連語義 $\mathcal{S}_s^U$，および異義 $\mathcal{S}_s^N$（§ 4.3.4）との類似度の平均値である．

$$
\begin{aligned}
\rho_{\mathcal{S}^P} &= \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{1}{|\mathcal{S}_s^P|}\sum_{s'\in\mathcal{S}_s^P}\rho_{s,s'},\\
\rho_{\mathcal{S}^U} &= \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{1}{|\mathcal{S}_s^U|}\sum_{s'\in\mathcal{S}_s^U}\rho_{s,s'},\\
\rho_{\mathcal{S}^N} &= \frac{1}{|\mathcal{S}^N|}\sum_{s\in\mathcal{S}^N}\frac{1}{|\mathcal{S}_s^N|}\sum_{s'\in\mathcal{S}_s^N}\rho_{s,s'},
\end{aligned}
\tag{4.17}
$$

ただし $\rho_{\mathcal{S}^N}$ については，$\mathcal{S}^N = \{s; |\mathcal{S}_s^N| > 0\}$，すなわち異義を 1 つ以上持つ語義のみを対象として平均を計算した．

$\rho_{\mathcal{W}^{\mathrm{gt}}}$ は，対象語 w と正解語義 $\mathcal{S}_w^{\mathrm{gt}}$ との平均類似度である．なお計算には本実験の評価データを再利用した．

$$
\rho_{\mathcal{W}^{\mathrm{gt}}} = \frac{1}{|\mathcal{W}|}\sum_{w\in\mathcal{W}}\frac{1}{|\mathcal{S}_w^{\mathrm{gt}}|}\sum_{s\in\mathcal{S}_w^{\mathrm{gt}}}\rho_{w,s}. \tag{4.18}
$$

提案手法による適応ずみ埋め込みの類似度と，適応前の BERT 埋め込みの類似度の違い（$\texttt{SS-WSD}_{\texttt{emb}}$ と BERT の差）に注目すると，無関連 $\rho_{\mathcal{S}^U}$ および異義 $\rho_{\mathcal{S}^N}$ は類似度が低下し，対象語と正解語義 $\rho_{\mathcal{W}^{\mathrm{gt}}}$ は類似度が上昇している．この結果は，意味ネットワークの構造に適応することによって，意味的に関連する語義対および語義・単語対を近付け，関連しない語義対および異義対を遠ざけるという狙いが，提案手法で実現されたことを示している．

それでは，語義対や語義・単語対を近付ける・遠ざける操作は，語義曖昧性解消に有効だといえるだろうか．これを考察するために，対象語と正解語義の類似度から無関連および異義の類似度を除算した類似度のマージン $\Delta\rho_{\mathcal{S}^U}$ および $\Delta\rho_{\mathcal{S}^N}$ を計算し，WSD タスクの精度と比較してみる．

$$
\begin{aligned}
\Delta\rho_{\mathcal{S}^U} &= \rho_{\mathcal{W}^{\mathrm{gt}}} - \rho_{\mathcal{S}^U}\\
\Delta\rho_{\mathcal{S}^N} &= \rho_{\mathcal{W}^{\mathrm{gt}}} - \rho_{\mathcal{S}^N}
\end{aligned}
\tag{4.19}
$$

類似度のマージンは，特定の語義からみたときに，語義とおなじ意味の用例

表 4.7: 語義対および対象語・正解語義対の平均 cosine 類似度．$\rho_{\mathcal{S}^P}$, $\rho_{\mathcal{S}^U}$, $\rho_{\mathcal{S}^N}$ はそれぞれ，関連・無関連・異義との類似度（式 4.17）．$\rho_{\mathcal{W}^{\mathrm{gt}}}$ は対象語と正解語義の類似度（式 4.18）．$\Delta\rho_*$ は $\rho_{\mathcal{W}^{\mathrm{gt}}}$ に対するマージン（式 4.19）．“WSD” 列は，表 4.5および表 4.6の “WSD (All)” 列を再掲．アブレーションの行は WSD 精度の降順に整列．

手法	$\rho_{\mathcal{S}^P}$	$\rho_{\mathcal{S}^U}$	$\rho_{\mathcal{S}^N}$	$\rho_{\mathcal{W}^{\mathrm{gt}}}$	$\Delta\rho_{\mathcal{S}^U}\uparrow$	$\Delta\rho_{\mathcal{S}^N}\uparrow$	WSD
BERT	0.91	0.77	0.87	0.64	−0.12	−0.23	65.2
SS-WSD$_{\mathrm{emb}}$	0.88	0.64	0.78	0.77	0.13	−0.01	74.9
– 異義 $\mathcal{S}^N$ との反発	0.87	0.61	0.79	0.77	0.17	−0.02	73.5
– 対象語埋め込みの適応	0.88	0.64	0.78	0.63	−0.01	−0.15	71.7
– 吸引・反発学習	0.92	0.79	0.90	0.81	0.02	−0.08	71.6
– 自己学習	0.88	0.64	0.78	0.61	−0.02	−0.17	70.5
– 無関連語義 $\mathcal{S}^U$ との反発	0.90	0.73	0.79	0.73	0.00	−0.06	69.9

が，同じ単語の異なる語義や無関連な語義よりも近くにあるかを示している．このことから，類似度のマージンが大きければ正解語義が最近傍になりやすいと考えられるため，語義曖昧性解消タスクへの適合性の目安となる可能性がある．アブレーション分析（§ 4.6.2）での各設定における類似度のマージンを計算し，WSD タスク精度の降順に整列した結果を表 4.7の 3 行目以降に示す．この結果からは，類似度のマージン $\Delta\rho_{\mathcal{S}^U}$ および $\Delta\rho_{\mathcal{S}^N}$ が大きいほど，おおむね WSD タスク精度が高くなる傾向がみられる．したがって，埋め込みを意味ネットワークに適応させることは，語義とおなじ意味の用例を異なる語義や無関連な語義よりも近づける効果によって，最近傍法の精度向上に有効であったと示唆される．またアブレーションの各設定における類似度のマージンを比較すると，自己学習は上述した効果への寄与が大きいことがわかる．このことから，自己学習における擬似正例の付与アルゴリズムを高度化することが，一層の精度向上に有効である可能性を指摘できる．

4.6.6 非一般的な語義に対する有効性の分析

教師あり手法による語義曖昧性解消は，本実験でも確認されたように，基本的には知識ベース手法の性能を上回る（§ 4.5）．しかし，教師あり手法における事実上標準の訓練データである SemCor の採録事例は，使用頻度が高い一般的な語義に偏っているとの指摘がある (Pasini 2020)．たとえば *mouse* の非典型的な語義 “臆病な人” の用例は，SemCor には含まれていない．このため，非一般的な語

義，具体的にはSemCorに非採録または非典型的な語義は，識別精度が低いことが報告されている (Maru et al. 2022)．これに対して知識ベース手法はSemCorに依存しないため，非一般的な語義では教師あり手法よりも有効かもかもしれない．なかでも提案手法は意味ネットワーク構造の特徴に対してBERT埋め込みを適応させることから，語義の一般性によらずどちらでも性能を発揮できる可能性がある．そこで本節では，非一般的な語義（たとえば *mouse* の“臆病な人”）の評価に適したベンチマーク用データセットを用いて，教師あり手法に対する知識ベース手法の優位性の有無，提案手法の性能，そして提案手法が予測する語義の傾向を分析する．

評価

本分析で使用する評価データセットは，WSD Hard Benchmark (Maru et al. 2022)である．このデータセットはALL_{NEW}，S10_{NEW}，42D，hardENおよびsoftENの5つのサブセットで構成されており（§ 2.3.2），本分析では特に42Dおよびhard ENに焦点を当てる．42Dは，正解語義が 1) SemCorでアノテーションされていない および 2) WordNet first senseではない というふたつの条件を満たす事例のみからなる．この特性により，42Dは非一般的な語義に対する有効性を調べるうえで有効である．またSemCorを訓練データとする教師あり手法においては，学習時に見ていない語義に対するゼロショット設定での評価とみなすことができる．hardENは，教師ありアプローチにおける主な既存手法いずれでも不正解となる事例のみからなる．この特性により，hardENは，教師ありアプローチが苦手とする事例に対する有効性を検討することに適している．

評価指標は，Maru et al. (2022) の推奨するマクロF値を用いる．提案手法の評価は，本実験と同様に，TaM経験則を使用しない場合と併用する場合の両方について報告する．

ベースライン

提案手法と比較するベースラインは，本実験（§ 4.4.3）とほぼ同一である．まず，教師ありアプローチに対しては`BEM` (Blevins and Zettlemoyer 2020) および`ESCHER` (Barba et al. 2021b) と比較する．`ESCHER`は42Dにおける最高精度の手法 (Maru et al. 2022)，`BEM`は本実験でもベースラインに選択した手法である．

つぎに，知識ベースアプローチに対しては，本実験と同様に BERT ，$\texttt{SREF}_{\texttt{emb}}$ ，および $\texttt{SREF}_{\texttt{kb}}$ と比較する．本実験で採用したベースラインのうち Sup-kNN および $\texttt{WN1}^{\texttt{st}}$ は，予測方法の原理上 42D のスコアがゼロになることが自明である[5] ため採用しなかった．また COE は，予測に必要な文書情報の定義が不明瞭であるため採用しなかった．

実験結果

表 4.8: WSD Hard Benchmark におけるサブセット別の WSD タスク精度．区分の Sup は教師あり手法，KB は知識ベース手法．提案手法の $\texttt{SS-WSD}_{\texttt{emb,kb}}$ 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．太字は全区分における最高精度．{ESCHER, BEM} はMaru et al. (2022) から引用．

区分	手法	ALL_{NEW}	S10_{NEW}	42D	hardEN	softEN
Sup	ESCHER	**78.7**	78.0	58.9	0.0	**83.7**
	BEM	75.6	77.1	53.2	0.0	80.3
KB	BERT	65.3	66.2	**65.4**	**43.5**	68.0
	$\texttt{SREF}_{\texttt{emb}}$	69.0	71.6	60.3	36.9	71.7
	$\texttt{SS-WSD}_{\texttt{emb}}$	72.9	**78.7**	63.3	30.9	76.9
		(0.4)	(1.4)	(1.4)	(1.3)	(0.6)
KB+TaM	$\texttt{SREF}_{\texttt{kb}}$	70.0	74.5	60.5	31.5	73.6
	$\texttt{SS-WSD}_{\texttt{kb}}$	74.7	77.6	58.2	28.9	77.8
		(0.5)	(1.9)	(1.4)	(1.3)	(0.7)

表 4.8に，ベースラインおよび提案手法の WSD タスク性能を示す．まず，42D および hardEN では，区分 KB に属するすべての知識ベース手法が教師あり手法を上回った．特に 42D では，BERT ，提案手法である $\texttt{SS-WSD}_{\texttt{emb}}$ ，$\texttt{SREF}_{\texttt{emb}}$ の順に性能が高く，BERT および $\texttt{SS-WSD}_{\texttt{emb}}$ は，最高精度の教師あり手法である ESCHER をそれぞれ 6.5 ポイントおよび 4.4 ポイント上回った．次に，非一般的な語義に限定しないサブセットである ALL_{NEW} では，原則として知識ベース手法が教師あり手法を下回った．これは本実験で報告した，Unified Evaluation Framework for WSD のデータセットによる評価と同一である．一方で S10_{NEW} においては，統計的に有意ではないものの $\texttt{SS-WSD}_{\texttt{emb}}$ が ESCHER を 0.7 ポイント上回った．S10_{NEW} は

[5] $\texttt{WN1}^{\texttt{st}}$ は WordNet first sense を予測する．また Sup-kNN は SemCor に含まれる語義または WordNet first sense を予測する．いずれも 42D では決して正解にならない．

特定の分野（生物多様性）からテキストを収集している (Agirre et al. 2010) ため，非一般的な語義が多いのかもしれない．

以上の結果から，ふたつの考察が得られる．ひとつめは，BERT 埋め込みを用いて対象語に最も近い語釈文を選ぶという知識ベース手法の方法論は，非一般的な語義が正解となる事例や，教師あり手法が苦手とする事例においては，教師あり手法よりも有効だということである．このことは，訓練データの偏りが予測語義に影響するという教師あり手法の課題に対して，知識ベース手法が解決の手がかりになる可能性を示している．ふたつめは，提案手法によって BERT が計算した埋め込みを適応させることは，非一般的な語義における性能が若干低下するかわりに，一般的な語義における性能が，分野によっては教師あり手法に匹敵するまでに改善するということである．このことは，埋め込みを適応させると，非一般的な語義にかわって一般的な語義が最近傍になる傾向があることを示唆している．

最後に，提案手法において TaM 経験則を併用した場合と使用しない場合の差異，すなわち $\texttt{SS-WSD}_{\texttt{kb}}$ と $\texttt{SS-WSD}_{\texttt{emb}}$ の差に注目すると，42D および hardEN では 2 から 5 ポイント低下する一方で，ALL_{NEW} では約 2 ポイント上昇している．したがって TaM 経験則は，非一般的な語義よりも一般的な語義を選好するように予測語義を変更する働きをすることが示唆される．

表 4.9: 42D の事例に対して提案手法 $\texttt{SS-WSD}_{\texttt{emb}}$ が予測した語義の特徴．予測語義が BERT と一致する事例と異なる事例に分割して分析．

予測語義が	事例数	正解率	予測語義が $\texttt{WN1}^{\texttt{st}}$ と一致	予測語義が SemCor に存在
BERT と一致	268	70.9	15.7%	17.2%
BERT と異なる	102	33.3	33.3%	50.0%

提案手法によって BERT が計算した埋め込みを適応させることは，非一般的な語義よりも一般的な語義を選好するように予測を変更する働きがあるのだろうか．これを検証するため，42D を構成する 370 件の事例を，$\texttt{SS-WSD}_{\texttt{emb}}$ の予測が BERT の予測と一致する事例と異なる事例のふたつに分割して，それぞれの予測語義の特徴を分析した．結果を表 4.9に示す．予測が BERT と異なる事例では，WordNet first sense または SemCor に存在する語義を予測する割合が，予測が BERT と一致する事例の 2 から 3 倍に増加している．たとえば SemCor に存在する語義を予測する割合は，17.2%から 50.0%へと約 30 ポイント増加している．こ

れは，適応ずみ埋め込みを用いた最近傍法では，適応前と比較すると一般的な語義が予測されやすくなることを支持する結果である．ただし本実験で報告したとおり，標準的な評価データセットでは，提案手法の SS-WSD$_{\text{emb}}$ が一般的な語義を強く選好する Sup-kNN や WN1$^{\text{st}}$ の性能を上回っている（表 4.5）．このことから，提案手法が常に一般的な語義を優先するのではなく，事例に応じて適切に振る舞っていることが示唆される．一方で，42D で BERT が SS-WSD$_{\text{emb}}$ を上回ることは，非一般的な語義に対して改善の余地が存在することを示唆している．具体的には，多様な単語や語義の用例を含むコーパスを，自己学習の訓練データとして用いることが有効かもしれない．訓練データ量の影響（§ 4.6.4）にて述べた対策とおなじく，大規模な平文コーパスから単語や語義の偏りを軽減するようにサンプリングすることで，そのようなコーパスが構築できる可能性がある．

4.7 本章のまとめ

本章では，BERT によって計算した語釈文の語義埋め込みおよび用例文の対象語埋め込みを，単語の語義および意味関係からなるネットワークの構造に適応させる手法を提案した．また，適応ずみ埋め込みを用いた最近傍法によって知識ベース語義曖昧性解消（WSD）を解いた．提案手法の核心は，ネットワーク構造における特徴を埋め込み間の類似度に反映させることで，関連する語義および語義と単語を近づけて，無関連な語義およびおなじ単語の異なる語義を遠ざけることである．これらの操作は，埋め込みの変換関数を吸引・反発学習および自己学習によって訓練することで実現される．各学習の訓練データは WordNet および平文コーパスであり，教師信号は意味ネットワーク上の単語と語義の配置における特徴から与えられる．具体的には，近付ける対象は隣接する語義対および語義・単語対，遠ざける対象は単語を介して 2 ホップで結ばれる語義対および，つながりのない語義対である．多義語に隣接する複数の語義の中でどの語義に近付けるかという課題は，自己学習，すなわち埋め込みの最近傍語義を近付けるべき擬似正例とみなすブートストラッピングで対処する．

提案手法の有効性を示すため，適応させた埋め込みを使用して対象語の最近傍語義を選ぶ方法により WSD タスクを解いた．その結果，提案手法は既存研究が用いた関連語義を互いに近付ける方法よりも約 4 ポイント高い精度を達成した．また，用例文に注釈された正解語義を用いて語義埋め込みを計算する単

純な教師あり手法をも上回った．適応による精度改善幅は，動詞で顕著だった．これは品詞ごとにネットワークの密度が異なることを反映している可能性がある．既存研究で提案された語義選択を再試行する経験則（Try-again Mechanism, § 4.3.5）を提案手法と併用するとさらに精度が向上し，推論時に文書情報が必要な従来手法を上回り，知識ベース WSD の最高精度を達成した．これにより，文書レベルの文脈情報に依存して汎用性を犠牲にせずとも，文単体のみでも埋め込みの意味適応によって高精度が実現できることを示した．

提案手法の有効性分析では，吸引・反発学習と自己学習の併用，対象語埋め込みの適応，および変換時の距離制約の効果が大きいことが示された．このことは，語義埋め込みだけでなく対象語埋め込みも適応させることの重要性を示している．距離制約については，制約の強度を変更した場合の精度の変化を観察することで有効性を実証するとともに，最近傍語義を正解と入れ替えるために必要な埋め込み類似度の変化は微小であることを観察して妥当性を示した．非一般的な語義を対象とした曖昧性解消の有効性分析では，最高精度の教師あり手法よりも約 4 ポイント高い精度を達成した．ただし適応させない埋め込みはさらに高い精度だった．このことは，訓練データにおける語義の偏りによって非一般的な語義の性能が犠牲になるという教師あり手法の課題に対して，提案手法のような埋め込みの最近傍法にもとづく知識ベース手法が貢献できる可能性を示している．

提案手法によって埋め込みが意味ネットワーク構造に適応したかを調べるため，語義対および語義・単語対の類似度を分析した．その結果，適応ずみの埋め込みでは，対象語と正解語義を近付け，同じ単語の異なる語義や，意味的に関連しない語義を遠ざけるという狙いどおりの類似度変化が生じていることを確認した．また，このような類似度の変化は自己学習の寄与が大きいこと，および変化が顕著であるほど WSD タスクの精度が向上する傾向を観察した．したがって，自己学習の強化はさらなる精度向上に寄与する可能性があるといえる．たとえば，擬似正例の付与アルゴリズムを高度化する方策が考慮に値するだろう．半教師あり学習の研究では，正例とその確信度を用いる手法が提案されている (Ishida et al. 2018)．単義語は語義曖昧性がないので誤りのない正解語義（正例）を付与できるという性質と組み合わせることで，より効果的なアルゴリズムの設計が可能かもしれない．自己学習に用いる平文コーパスのデータ量を減らした場合の分析からは，コーパスの単純な大規模化では性能向上は見込めな

いことが示された．だが単純な大規模化ではなく，さまざまな単語や語義をバランス良く網羅するように用例を抽出する方策と組み合わせるならば有効かもしれない．また，そのようなコーパスを用いた自己学習は，非一般的な語義に対する性能改善にも寄与する可能性がある．

第 5 章

結論

本研究では，fastText や BERT といった深層学習によって大規模言語データから学習した単語埋め込みを，WordNet のような人間が構築した語彙資源の知識に適応させることで，単語の意味を扱う問題での性能改善に取り組んだ．特に本研究は，語彙資源に直接記述されている単語の意味や語義間の意味関係だけでなく，その背景にあるデータ構造の特徴に焦点を当て，これに適応させるアプローチを採用した．すなわち，上位概念の特徴や性質を下位概念が継承することによる概念階層および，単語がとりうる語義と語義どうしの意味関係による意味ネットワークである．単語の意味を扱う問題は，文脈から独立した静的な意味を問うものと，文脈に依存して変化する多義性を問うものに大別されるが，本研究では前述したアプローチのもとで両方の問題に取り組んだ．具体的には，静的単語埋め込みの概念階層への適応による単語間の上位下位関係識別および，文脈依存単語埋め込みにもとづく対象語および語義埋め込みの意味ネットワークへの適応による語義曖昧性解消のふたつの研究を実施した．

概念階層への適応による上位下位関係識別においては，静的単語埋め込みを階層コード表現に変換することで，概念階層の特徴である推移律および反対称律に沿うような推論をする手法を提案した．階層コード表現への変換は，入力した単語埋め込みを，上位下位語ペアに対しては包含関係にあるコード，非上位下位語ペアに対しては包含関係にないコードを出力するようにモデルを訓練することで実現する．ただし離散表現のコード対における包含関係の判定は「あり」か「なし」かの二値であり，最適化に必要なモデルの勾配が計算できない．この課題に対して提案手法では，コード表現を連続緩和すること，および連続緩和

したコードを分布パラメータとするコードの確率分布を考えることで，コード間の包含関係の期待値という連続変数として計量するという解決策を考案した．変換により得られた単語のコード表現どうしの包含関係を用いて上位下位関係を推論したところ，一部のタスクでは先行研究を上回る性能を達成した．タスクの誤り分析によって，コードが概念の抽象度を捉えている可能性や，上位下位関係を全体部分関係と誤認する傾向があることを示した．また，離散化したコードの割り当てを単語のクラスタリングとみなして分析すると，WordNet の概念階層の特徴と類似した割り当てがなされていることを確認した．

意味ネットワークへの適応による語義曖昧性解消においては，文脈依存埋め込みから計算した対象語および語義の埋め込みを変換して，ネットワーク構造の特徴に合うように単語と語義および語義どうしの類似度を変更する手法を提案した．具体的には，隣接する語義対および語義・単語対は近付けながら，単語を介して 2 ホップで結ばれる語義対および，つながりのない語義対は遠ざけるように変換するモデルを訓練することで実現する．ただし多義語は複数の語義と隣接するので，単語が出現した文脈に合致する語義を選んで近付けたい．この課題に対して提案手法では，吸引・反発学習および自己学習を併用することで，語義どうしの類似度，対象語と語義の類似度を同時に変更するという解決策を考案した．変換により適応させた埋め込みを用いて，対象語にもっとも近い語義を選択する方法で正解語義を推論したところ，知識ベース語義曖昧性解消タスクで先行研究を上回る性能を達成した．タスクの有効性分析からは，先行研究のように語義埋め込みの適応だけでなく，対象語埋め込みも適応させることの重要性が示された．また，適応ずみの埋め込みによる類似度を分析すると，対象語と正解語義を近付け，同じ単語の異なる語義や，つながりのない語義を遠ざけるという，ネットワーク構造の特徴に合致した類似度の変化が生じていることを確認した．さらに，このような類似度の変化が顕著であるほど WSD タスクの精度が向上している傾向を観察した．

これらの結果は，単語の意味を表現するふたつの方法論である，統計に基づく単語埋め込みと人間の知識に基づく語彙資源を統合する手段として，学習ずみの埋め込みを意味関係の背景にある階層構造やネットワーク構造の特徴に適応させることが，上位下位関係識別や語義曖昧性解消の性能改善に寄与することを示している．以上の貢献は，性能改善という工学的な利点だけでなく，深層学習と人間の知識それぞれの長所を相互補完するアプローチの可能性を示すも

のでもある．すなわち，大規模言語データから学習することによるスケーラビリティおよび計算機上での操作性と，人間が記述・理解できる自然言語の表現がもたらす緻密性および解釈可能性，これら双方の利点を活用するアプローチへの一助になると考えている．

本研究を発展させる今後の展望として，語彙資源の構築，多言語化，および大規模言語モデルへの語彙資源の統合が挙げられる．

語彙資源の構築においては，タクソノミの自動構築（Automatic Taxonomy Construction: Bordea et al. (2016, 2015), Buitelaar et al. (2005)）および自動拡張（Automatic Taxonomy Expansion: Shen et al. (2020), Jurgens and Pilehvar (2016)）が重要な課題である．これらのタスクの目標は，言語データからタクソノミ（上位下位関係による階層構造）を構築または拡張することであり，広義には語彙資源の自動生成や維持を目的としている．一方でタクソノミの構築は，単語間の上位下位関係識別だけでなく，識別した上位下位の集合を閉路や非連結などの異常がない階層構造に組み立てる必要があるため，難易度が高い．概念階層への適応による上位下位関係識別で提案した手法は，各単語に対して階層コードを割り当てることができるため，タクソノミ構築に応用しやすい可能性がある．

多言語化においては，多言語語義曖昧性解消（Multilingual Word Sense Disambiguation: Navigli et al. (2013)）が重要な課題である．多言語 WSD は，言語間の技術格差の解消に加えて，対訳ペアの意味の一貫性を調べることによる翻訳誤りの検証にも寄与する (Campolungo et al. 2022)．意味ネットワークへの適応による語義曖昧性解消で提案した手法は語彙資源のみを活用するため，多言語化が容易である．具体的には，多言語版語彙資源である BabelNet（§ 2.3.2）(Navigli et al. 2021) および，多言語テキストで訓練された言語モデル (Feng et al. 2022, Conneau et al. 2020) を提案手法に組み合わせることができる．また多言語埋め込みに対する意味適応の有効性という観点からは，英語資源のみを用いて適応させるゼロショットの言語間転移学習や，異なる言語で記述された語釈文や用例のペアによる訓練の効果などは興味深い研究課題である．

大規模言語モデルへの語彙資源の統合は，モデルに単語の意味や意味関係の知識を教えることを目的とする課題である．本研究ではモデルが計算した埋め込みを語彙知識に適応させて推論に用いることの有効性を示した．これに対して本課題では，言語モデルの内部状態を操作して出力文に語彙知識を反映させ

ることを目的とする．たとえば「デンファレはどこで買えますか」という問いに「デンファレとはなんですか？」ではなく「花屋で買えます」と回答させることである．既存研究によると，大規模言語モデルにはテキストから学習した百科事典的・常識的知識が格納されていること (Petroni et al. 2019) や，タスクの説明文を与えるだけで任意のタスクが解けること (Brown et al. 2020) が報告されている．また，間違いや古くなった知識を修正したいという動機から，モデルパラメータの局所的な更新や隠れ状態ベクトルへの干渉によって内部状態を操作することにより，格納ずみの知識を編集する（例：エッフェル塔はパリではなくローマにある，と言わせる）手法が提案されている (Meng et al. 2022, Dai et al. 2022)．大規模言語モデルに指示することで任意の問題を解くという方法論のすぐれた汎用性を考慮すると，内部状態の操作によってモデルに語彙知識を教える手法を開発することは，機械翻訳や質問応答など幅広いタスクに直接貢献しうる重要な研究課題だといえる．

業績リスト

ジャーナル論文

1. 水木 栄, 岡崎 直観. 階層コード表現学習による上位下位関係の識別. 情報処理学会論文誌データベース (TOD), 14(4):8–23, 2021 年 10 月.

国際学会発表

1. Sakae Mizuki and Naoaki Okazaki. Semantic Specialization for Knowledge-based Word Sense Disambiguation. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL, pp. 3457-3470, May 2023.

2. Sakae Mizuki and Naoaki Okazaki. Analyzing the Variation Property of Contextualized Word Representations. In AI 2019: Advances in Artificial Intelligence, pp. 393–405, December 2019.

国内口頭発表

1. 水木 栄, 岡崎 直観. 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. 言語処理学会第 29 回年次大会 (NLP2023), pp. 622-627, 2023 年 3 月. 言語処理学会第 29 回年次大会優秀賞.

2. 石川 遼伍, 丹羽 彩奈, 水木 栄, 岡崎 直観. 疑似訓練データによる格助詞の省略に頑健な係り受け解析. 言語処理学会第 28 回年次大会 (NLP2022), pp. 1808-1813, 2022 年 3 月.

3. 水木 栄, 岡崎 直観. 階層コード表現を用いた上位下位関係の識別. 言語処理学会第 27 回年次大会 (NLP2021), pp. 1236-1241, 2021 年 3 月. 言語処理学会第 27 回年次大会優秀賞.

寄稿および講演

1. 水木 栄. 「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」ができるまで. 自然言語処理, vol. 30, no. 3, pp. 1105-1109, 2023 年 9 月.

2. 水木 栄. 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. NLP コロキウム, 2023 年 5 月.

参考文献

Eneko Agirre, Oier Lopez de Lacalle, Christiane Fellbaum, Shu-Kai Hsieh, Maurizio Tesconi, Monica Monachini, Piek Vossen, and Roxanne Segers (2010) "SemEval-2010 Task 17: All-Words Word Sense Disambiguation on a Specific Domain," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 75–80.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019) "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631.

Carl Allen and Timothy M. Hospedales (2019) "Analogies Explained: Towards Understanding Word Embeddings," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 223–231.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski (2016) "A Latent Variable Model Approach to PMI-based Word Embeddings," *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 385–399.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma (2017) "A Simple but Tough-to-Beat Baseline for Sentence Embeddings," in *5th International Conference on Learning Representations*.

Ignacio Arroyo-Fernández, Carlos-Francisco Méndez-Cruz, Gerardo Sierra, Juan-Manuel Torres-Moreno, and Grigori Sidorov (2019) "Unsupervised sen-

tence representations as word information series: Revisiting TF-IDF," *Comput. Speech Lang.*, Vol. 56, pp. 107–129.

Ben Athiwaratkun and Andrew Gordon Wilson (2018) "Hierarchical Density Order Embeddings," in *Proceedings of the 6th International Conference on Learning Representations*.

Amir Bakarov (2018) "A Survey of Word Embeddings Evaluation Methods," *CoRR*, Vol. abs/1801.09536.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales (2019) "TuckER: Tensor Factorization for Knowledge Graph Completion," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 5184–5193.

Edoardo Barba, Luigi Procopio, and Roberto Navigli (2021a) "ConSeC: Word Sense Disambiguation as Continuous Sense Comprehension," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1492–1503.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli (2021b) "ESC: Redesigning WSD with Extractive Sense Comprehension," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4661–4672: Association for Computational Linguistics.

Marco Baroni and Alessandro Lenci (2011) "How we BLESSed distributional semantic evaluation," in *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pp. 1–10.

Gábor Berend (2023) "Masked Latent Semantic Modeling: an Efficient Pretraining Alternative to Masked Language Modeling," in *Findings of the Association for Computational Linguistics*, pp. 13949–13962.

Michele Bevilacqua and Roberto Navigli (2020) "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.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli (2021) "Recent Trends in Word Sense Disambiguation: A Survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 4330–4338.

Or Biran and Kathleen R. McKeown (2013) "Classifying Taxonomic Relations between Pairs of Wikipedia Articles," in *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pp. 788–794.

Terra Blevins and Luke Zettlemoyer (2020) "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.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov (2017) "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146.

Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli (2015) "SemEval-2015 Task 17: Taxonomy Extraction Evaluation (TExEval)," in *Proceedings of the 9th International Workshop on Semantic Evaluation*, pp. 902–910.

Georgeta Bordea, Els Lefever, and Paul Buitelaar (2016) "SemEval-2016 Task 13: Taxonomy Extraction Evaluation (TExEval-2)," in *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 1081–1091.

Tom B. Brown, Benjamin Mann, Nick Ryder et al. (2020) "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*

33: Annual Conference on Neural Information Processing Systems 2020, pp. 1877–1901.

Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini (2005) *Ontology Learning from Text: Methods, Evaluation and Applications*, Vol. 123: IOS Press.

José Camacho-Collados (2017) "Why we have switched from building full-fledged taxonomies to simply detecting hypernymy relations," *CoRR*, Vol. abs/1703.04178.

José Camacho-Collados and Mohammad Taher Pilehvar (2018) "From Word To Sense Embeddings: A Survey on Vector Representations of Meaning," *Journal of Artificial Intelligence Research*, Vol. 63, pp. 743–788.

Niccolò Campolungo, Federico Martelli, Francesco Saina, and Roberto Navigli (2022) "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.

Steven Cao, Nikita Kitaev, and Dan Klein (2020) "Multilingual Alignment of Contextual Word Representations," in *8th International Conference on Learning Representations*.

Qianglong Chen, Feng-Lin Li, Guohai Xu, Ming Yan, Ji Zhang, and Yin Zhang (2022) "DictBERT: Dictionary Description Knowledge Enhanced Language Model Pre-training via Contrastive Learning," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 4086–4092.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton (2020) "A Simple Framework for Contrastive Learning of Visual Representations," in *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119, pp. 1597–1607.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal et al. (2020) "Unsupervised Cross-lingual Representation Learning at Scale," in *Proceedings of the*

58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto (2013) *Recognizing Textual Entailment: Models and Applications*, Synthesis Lectures on Human Language Technologies: Morgan & Claypool Publishers.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei (2022) "Knowledge Neurons in Pretrained Transformers," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 8493–8502.

Sarthak Dash, Md. Faisal Mahbub Chowdhury, Alfio Gliozzo, Nandana Mihindukulasooriya, and Nicolas Rodolfo Fauceglia (2020) "Hypernym Detection Using Strict Partial Order Networks," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 7626–7633.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou (2019) "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699.

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*, pp. 4171–4186.

Philip Edmonds and Scott Cotton (2001) "SENSEVAL-2: Overview," in *Proceedings of Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pp. 1–5.

Kawin Ethayarajh (2019) "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 55–65.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith (2015) "Retrofitting Word Vectors to Semantic Lexicons," in *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1606–1615.

Christiane Fellbaum (1998) *WordNet: An Electronic Lexical Database*: The MIT Press.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang (2022) "Language-agnostic BERT Sentence Embedding," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 878–891.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin (2001) "Placing search in context: the concept revisited," in *Proceedings of the Tenth International World Wide Web Conference*, pp. 406–414.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur (2020) "Sharpness-Aware Minimization for Efficiently Improving Generalization," *CoRR*, Vol. abs/2010.01412.

William A. Gale, Kenneth Ward Church, and David Yarowsky (1992) "One Sense Per Discourse," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, USA, February 23-26, 1992*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen (2021) "SimCSE: Simple Contrastive Learning of Sentence Embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910.

Artur S. d'Avila Garcez and Luís C. Lamb (2020) "Neurosymbolic AI: The 3rd Wave," *CoRR*, Vol. abs/2012.05876.

John M. Giorgi, Osvald Nitski, Bo Wang, and Gary D. Bader (2021) "DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations," in

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing,, pp. 879–895.

Marti A. Hearst (1992) "Automatic Acquisition of Hyponyms from Large Text Corpora," in *14th International Conference on Computational Linguistics*, pp. 539–545.

Felix Hill, Roi Reichart, and Anna Korhonen (2015) "SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation," *Computational Linguistics*, Vol. 41, No. 4, pp. 665–695.

Jeremy Howard and Sebastian Ruder (2018) "Universal Language Model Fine-tuning for Text Classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 328–339.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama (2017) "Learning Discrete Representations via Information Maximizing Self-Augmented Training," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 1558–1567.

Lawrence Hubert and Phipps Arabie (1985) "Comparing partitions," *Journal of classification*, Vol. 2, No. 1, pp. 193–218.

Aliaksandr Huminski and Hao Zhang (2018) "WordNet Troponymy and Extraction of "Manner-Result" Relations," in *Proceedings of the 9th Global Wordnet Conference*, pp. 407–411.

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

Takashi Ishida, Gang Niu, and Masashi Sugiyama (2018) "Binary Classification from Positive-Confidence Data," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018*, pp. 5921–5932.

Daniel Jurafsky and James H. Martin (2009) *Speech and Language Processing (2nd Edition)*, Chap. 20, pp. 678–679: Prentice-Hall, Inc.

David Jurgens and Mohammad Taher Pilehvar (2016) "SemEval-2016 Task 14: Semantic Taxonomy Enrichment," in *Proceedings of the 10th International Workshop on Semantic Evaluation*, pp. 1092–1102.

Douwe Kiela, Laura Rimell, Ivan Vulic, and Stephen Clark (2015) "Exploiting Image Generality for Lexical Entailment Detection," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pp. 119–124.

Diederik P. Kingma and Max Welling (2014) "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations*.

Maximilian Köper, Christian Scheible, and Sabine Schulte im Walde (2015) "Multilingual Reliability and "Semantic" Structure of Continuous Word Spaces," in *Proceedings of the 11th International Conference on Computational Semantics*, pp. 40–45.

Caterina Lacerra, Michele Bevilacqua, Tommaso Pasini, and Roberto Navigli (2020) "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.

Benjamin J. Lengerich, Andrew L. Maas, and Christopher Potts (2018) "Retrofitting Distributional Embeddings to Knowledge Graphs with Functional Relations," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2423–2436.

Michael Lesk (1986) "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *Proceed-*

ings of the 5th Annual International Conference on Systems Documentation, SIGDOC 1986, pp. 24–26.

Yoav Levine, Barak Lenz, Or Dagan et al. (2020) "SenseBERT: Driving Some Sense into BERT," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4656–4667.

Omer Levy and Yoav Goldberg (2014) "Dependency-Based Word Embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 302–308.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan (2015) "Do Supervised Distributional Methods Really Learn Lexical Inference Relations?" in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 970–976.

Xiang Li, Luke Vilnis, Dongxu Zhang, Michael Boratko, and Andrew McCallum (2019) "Smoothing the Geometry of Probabilistic Box Embeddings," in *Proceedings of the 7th International Conference on Learning Representations*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith (2019) "Linguistic Knowledge and Transferability of Contextual Representations," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1073–1094.

Yinhan Liu, Myle Ott, Naman Goyal et al. (2019) "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *CoRR*, Vol. abs/1907.11692.

Daniel Loureiro and Alípio Jorge (2019) "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.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and José Camacho-Collados (2021) "Analysis and Evaluation of Language Models for Word Sense Disambiguation," *Comput. Linguistics*, Vol. 47, No. 2, pp. 387–443.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh (2017) "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables," in *5th International Conference on Learning Representations*.

Marco Maru, Simone Conia, Michele Bevilacqua, and Roberto Navigli (2022) "Nibbling at the Hard Core of Word Sense Disambiguation," in Smaranda Muresan, Preslav Nakov, and Aline Villavicencio eds. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 4724–4737.

Yoshihiro Maruyama (2021) "Symbolic and statistical theories of cognition: towards integrated artificial intelligence," in *Software Engineering and Formal Methods. SEFM 2020 Collocated Workshops: ASYDE, CIFMA, and CoSim-CPS*, pp. 129–146.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov (2022) "Locating and Editing Factual Associations in GPT," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*, pp. 17359–17372.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean (2013a) "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pp. 3111–3119.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean (2013b) "Efficient Estimation of Word Representations in Vector Space," in *1st International Conference on Learning Representations*.

Tomás Mikolov, Wen-tau Yih, and Geoffrey Zweig (2013c) "Linguistic Regularities in Continuous Space Word Representations," in *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*, pp. 746–751.

Tomás Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin (2018) "Advances in Pre-Training Distributed Word Representations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker (1993) "A Semantic Concordance," in *Human Language Technology: Proceedings of a Workshop Held at Plainsboro*.

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*, pp. 288–297.

Nikola Mrksic, Ivan Vulic, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gasic, Anna Korhonen, and Steve J. Young (2017) "Semantic Specialization of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints," *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 309–324.

Roberto Navigli (2009) "Word sense disambiguation: A survey," *ACM computing surveys (CSUR)*, Vol. 41, No. 2, pp. 10:1–10:69.

Roberto Navigli, David Jurgens, and Daniele Vannella (2013) "SemEval-2013 Task 12: Multilingual Word Sense Disambiguation," in *Proceedings of the 7th International Workshop on Semantic Evaluation*, pp. 222–231.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi (2021) "Ten Years of BabelNet: A Survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 4559–4567.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu (2016) "Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 454–459.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu (2017) "Hierarchical Embeddings for Hypernymy Detection and Directionality," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 233–243.

Maximilian Nickel and Douwe Kiela (2017) "Poincaré Embeddings for Learning Hierarchical Representations," in *Advances in Neural Information Processing Systems 30*, pp. 6338–6347.

Masataka Ono, Makoto Miwa, and Yutaka Sasaki (2015) "Word Embedding-based Antonym Detection using Thesauri and Distributional Information," in *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 984–989.

Tommaso Pasini (2020) "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.

Tommaso Pasini, Alessandro Raganato, and Roberto Navigli (2021) "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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018) "Deep Contextualized Word Representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2227–2237.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith (2019) "Knowledge Enhanced Contextual 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*, pp. 43–54.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller (2019) "Language Models as Knowledge Bases?" in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2463–2473.

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 4th International Workshop on Semantic Evaluations*, pp. 87–92.

Alexandre Rademaker, Bruno Cuconato, Alessandra Cid, Alexandre Tessarollo, and Henrique Andrade (2019) "Completing the Princeton Annotated Gloss Corpus Project," in *Proceedings of the 10th Global Wordnet Conference*, pp. 378–386.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever et al. (2018) "Improving language understanding by generative pre-training."

Alessandro Raganato, José Camacho-Collados, and Roberto Navigli (2017) "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.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim (2019) "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.

Nils Reimers and Iryna Gurevych (2019) "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3980–3990.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky (2020) "A Primer in BERTology: What We Know About How BERT Works," *Transactions of the Association for Computational Linguistics*, Vol. 8, pp. 842–866.

Stephen Roller, Douwe Kiela, and Maximilian Nickel (2018) "Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 358–363.

Dinghan Shen, Qinliang Su, Paidamoyo Chapfuwa, Wenlin Wang, Guoyin Wang, Ricardo Henao, and Lawrence Carin (2018) "NASH: Toward End-to-End Neural Architecture for Generative Semantic Hashing," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 2041–2050.

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han (2020) "TaxoExpan: Self-supervised Taxonomy Expansion with Position-Enhanced Graph Neural Network," in *The Web Conference 2020*, pp. 486–497.

Raphael Shu and Hideki Nakayama (2018) "Compressing Word Embeddings via Deep Compositional Code Learning," in *Proceedings of the 6th International Conference on Learning Representations*.

Benjamin Snyder and Martha Palmer (2004) "The English all-words task," in *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Chiraag Sumanth and Diana Inkpen (2015) "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, pp. 115–121.

Ian Tenney, Patrick Xia, Berlin Chen et al. (2019) "What do you learn from context? Probing for sentence structure in contextualized word representations," in *7th International Conference on Learning Representations*.

Shubham Toshniwal, Haoyue Shi, Bowen Shi, Lingyu Gao, Karen Livescu, and Kevin Gimpel (2020) "A Cross-Task Analysis of Text Span Representations," in *Proceedings of the 5th Workshop on Representation Learning for NLP*, pp. 166–176.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017) "Attention is All you Need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 5998–6008.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun (2016) "Order-Embeddings of Images and Language," in *4th International Conference on Learning Representations, Conference Track Proceedings*.

Elena Voita, Rico Sennrich, and Ivan Titov (2019) "The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4395–4405.

Ivan Vulic and Nikola Mrksic (2018) "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.

Ivan Vulic, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen (2017) "HyperLex: A Large-Scale Evaluation of Graded Lexical Entailment," *Computational Linguistics*, Vol. 43, No. 4, pp. 781–835.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen (2020) "Probing Pretrained Language Models for Lexical Semantics," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 7222–7240.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2018) "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355.

Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C-C Jay Kuo (2019) "Evaluating word embedding models: Methods and experimental results," *APSIPA transactions on signal and information processing*, Vol. 8, p. e19.

Dong Wang, Ning Ding, Piji Li, and Haitao Zheng (2021) "CLINE: Contrastive Learning with Semantic Negative Examples for Natural Language Understanding," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 2332–2342.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu (2018) "CosFace: Large Margin Cosine Loss for Deep Face Recognition," in *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274.

Ming Wang and Yinglin Wang (2020) "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.

Ming Wang and Yinglin Wang (2021) "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.

Ming Wang, Jianzhang Zhang, and Yinglin Wang (2021) "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.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang (2021) "KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation," *Transactions of the Association for Computational Linguistics*, Vol. 9, pp. 176–194.

Yinglin Wang, Ming Wang, and Hamido Fujita (2020) "Word Sense Disambiguation: A comprehensive knowledge exploitation framework," *Knowledge Based System*, Vol. 190, p. 105030.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David J. Weir, and Bill Keller (2014) "Learning to Distinguish Hypernyms and Co-Hyponyms," in *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 2249–2259.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann (2019) "Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings," in *Proceedings of the 15th Conference on Natural Language Processing*.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu (2015) "From Paraphrase Database to Compositional Paraphrase Model and Back," *Transactions of the Association for Computational Linguistics*, Vol. 3, pp. 345–358.

Zhibiao Wu and Martha Palmer (1994) "Verb Semantics and Lexical Selection," in *32nd Annual Meeting of the Association for Computational Linguistics*, pp. 133–138.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang (2022) “Dict-BERT: Enhancing Language Model Pre-training with Dictionary,” in *Findings of the Association for Computational Linguistics*, pp. 1907–1918.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu (2019) “ERNIE: Enhanced Language Representation with Informative Entities,” in *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pp. 1441–1451.

Lin Zheng, Qinliang Su, Dinghan Shen, and Changyou Chen (2020) “Generative Semantic Hashing Enhanced via Boltzmann Machines,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 777–788.

Zhi Zhong and Hwee Tou Ng (2012) “Word Sense Disambiguation Improves Information Retrieval,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 273–282.

岡崎直観・鶴岡慶雅・荒瀬由紀・宮尾祐介・鈴木潤 (2022) 『IT Text 自然言語処理の基礎』，オーム社.

隅田飛鳥・吉永直樹・鳥澤健太郎 (2009) 「Wikipedia の記事構造からの上位下位関係抽出」,『自然言語処理』，第 16 巻，第 3 号，3–24 頁.

国立国語研究所 (2004) 『分類語彙表 -増補改訂版-』，大日本図書.

水木栄 (2023) 「「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」ができるまで」,『自然言語処理』，第 30 巻，第 3 号，1105–1109 頁.

柏野和佳子 (2006) 「『分類語彙表』 の特徴と位置付け」,『日本語科学』，第 19 巻，143–160 頁.

林良彦 (2010) 「言語的オントロジーの構築と展開 (特集 オントロジーの進化と普及 (前編))」,『人工知能』，第 25 巻，第 3 号，335–344 頁.