論文

概念辞書の類義語と分散表現を利用した教師なし all-words WSD

鈴木　類[†]・古宮嘉那子[††]・浅原　正幸[†††]・佐々木　稔[††]・新納　浩幸[††]

all-words 語義曖昧性解消（以下 all-words WSD (word sense disambiguation)）とは文書中のすべての単語の語義ラベルを付与するタスクである．単語の語義は文脈，すなわち周辺の単語によって推定でき，周辺の単語同士が類似している場合中心の単語同士の語義も類似していると考える．そこで本研究では，対象単語とその類義語群から周辺単語の分散表現を作成し，ユークリッド距離を計算することで対象単語の語義を予測した．また，語義の予測結果をもとにコーパスを語義ラベル列に変換し，語義の分散表現を作成した．語義の分散表現を用いて周辺単語ベクトルを作成し直し，再び語義の予測を行った．コーパスには分類語彙表番号がアノテーションされた『現代日本語書き言葉均衡コーパス』(BCCWJ) を利用した．本研究では分類語彙表における分類番号を語義とし，類義語も分類語彙表から取得した．本研究では，提案手法とランダムベースライン，Pseudo Most Frequent Sense (PMFS)，Yarowsky の手法，LDAWN を比較し，提案手法が勝ることを示した．

キーワード：all-words，語義曖昧性解消，分類語彙表，分散表現

Unsupervised All-words WSD Using Synonyms and Embeddings

Rui Suzuki[†], Kanako Komiya[††], Masayuki Asahara[†††], Minoru Sasaki[††] and Hiroyuki Shinnou[††]

All-words word-sense disambiguation (all-words WSD) involves identifying the senses of all words in a document. Since a word's sense depends on the context, such as surrounding words, similar words are believed to have similar sets of surrounding words. Therefore, we predict target word senses by calculating Euclidean distances between the target words' surrounding word vectors and their synonyms using word embeddings. In addition, we replace word tokens in the corpus with their concept tags, that is, article numbers of the Word List by Semantic Principles using prediction results. After that, we create concept embeddings with the concept tag sequence and predict the senses of the target words using the distances between surrounding word

[†] 茨城大学大学院理工学研究科情報工学専攻, Major in Computer and Information Sciences, Graduate School of Science and Engineering, Ibaraki University
[††] 茨城大学大学院理工学研究科情報科学領域, Department of Computer and Information Sciences, Graduate School of Science and Engineering, Ibaraki University
[†††] 人間文化研究機構国立国語研究所, National Institute for Japanese Language and Linguistics

vectors, which consist the word and concept embeddings. This paper shows that concept embedding improved the performance of Japanese All-words WSD.

Key Words: *All-words, Word Sense Disambiguation, Word List by Semantic Principles*

1 はじめに

語義曖昧性解消（以下，WSD）とは多義語の語義ラベルを付与するタスクである．長年，英語のみならず日本語を対象とした WSD の研究が盛んに行われてきた．しかし，その多くは教師あり学習による対象単語を頻出単語に限定した WSD (lexical sample WSD) であるため，実用性が高いとは言えない．これに対し，文書中のすべての単語を対象とする WSD を all-words WSD という．all-words WSD のツールがあれば，より下流の処理の入力として，例えば品詞情報のように語義を利用することが可能になり，より実用的になると期待される．all-words WSD は，lexical sample WSD と異なり，教師ありの機械学習に利用する十分な量の訓練事例を得ることが難しいため，辞書などの外部の知識を利用して，教師なしの手法で行われることが一般的である．

all-words WSD の研究は日本語においては研究例が少ない．その理由のひとつに，all-words WSD を実行・評価するのに足りるサイズの語義つきコーパスがないことがあげられる．日本語の教師あり手法による WSD では，岩波国語辞典の語義が付与されている『現代日本語書き言葉均衡コーパス』（以下，BCCWJ）(Okumura, Shirai, Komiya, and Yokono 2011) がよく用いられてきた．しかし，知識ベースの手法で all-words WSD を行う場合に多用される類義語の情報は岩波国語辞典のような語義列記型の辞書からは得ることができない．英語の all-words WSD においては，WordNet[1]というシソーラスが辞書として主に利用されている．WordNet には日本語版も存在するが，基本的には英語版を和訳したものであり，日本語にしかない品詞の単語はどうするのかなどの問題点が残る．そのため，現在 BCCWJ に分類語彙表の意味情報がアノテーションされ，語義付きコーパスが整備されつつある．本研究では，整備されつつあるこのコーパス (Kato, Asahara, and Yamazaki 2018) を用いて，日本語を対象とした教師なし all-words WSD を行う．

分類語彙表とは単語を意味によって分類したシソーラスである．レコード総数は約 10 万件で，各レコードは類・部門・中項目・分類項目を表す“分類番号”によって分類されている．その他にも分類語彙表では“段落番号”，“小段落番号”，“語番号”が各レコードに割り振られており，それらすべての番号によってレコードが一意に決まるようになっている．また，分類語彙表には「意味的区切り」が 240 箇所に存在し，分類番号による分類をさらに細かく分けて

[1] https://wordnet.princeton.edu/

いる．本稿では分類語彙表から得られる類義語の情報を利用し，分類番号を語義とした日本語 all-words WSD の手法を提案する．

2 関連研究

WSD の手法は，大きく教師あり学習と知識ベース（教師なしの手法）の二つに分けることができる．一般的に，WSD を教師あり学習を用いた手法で行った場合，教師なしの手法に比べて高い精度を得ることができる．しかしその反面，十分な量の教師データ，すなわちタグ付きの用例文が必要なためその作成にコストがかかるという問題点がある．一方，教師なしの場合は教師データを必要としないためコストはかからないが，教師ありの手法と同等の精度を得ることは難しい．

WSD においては，一般に対象単語の文脈を素性とする．例えば，Yarowsky (1995) は，同一の連語や文書内では出現する単語に対する語義割り当てが一意であるという仮説 (Gale, Church, and Yarowsky 1992) のもと，教師なしによる WSD で高い精度を達成している．また，近年，文脈として，WSD の対象単語の周辺単語を分散表現で表す研究が行われている．Sugawara, Takamura, Sasano, and Okumura (2015) では，教師あり学習において対象単語の前後 N 単語ずつの単語の分散表現を基本的な素性として用い，その有効性を明らかにした．さらに，Yamaki, Shinnou, Komiya, and Sasaki (2016) は，単語の位置を規定しない・自立語以外の語を考慮しない，などして Sugawara らの手法を改善し成果を上げている．このように，対象単語を決めるうえで周辺の単語が大きな手掛かりとなることが知られている．そのため，本研究では，教師なしの手法を利用する際にも，周辺の単語の分散表現を手掛かりとして利用する．

一方で，本研究には階層的な概念辞書である，シソーラスも同時に利用する．WSD に分類語彙表などのシソーラスを用いる手法は数多く提案されている．特に，教師あり手法ではシソーラスから得られる単語の情報や上位概念を素性として利用することが多い．新納，佐々木，古宮 (2015) では，教師あり手法による WSD において分類語彙表などのシソーラスを素性に利用することの有効性や，上位概念のレベル（シソーラスの粒度）による精度の差があまりないことなどが報告されている．また，シソーラスを用いた辞書ベースの手法は，教師なしの手法のうち最も一般的な手法の一つである．Yarowsky (1992) はロジェのシソーラスを用いた教師なし手法による WSD の手法を提案した．また，小林，白井 (2018) はシソーラスを分類語彙表に置き換え，Yarowsky の手法に改良を加えた手法を提案している．これらの手法では，シソーラスにおいて対象単語の語義と同じ分類に属する単語の用例を集め，用例に出現しやすい自立語すなわち語義の特徴の重みを計算することで語義を予測している．また，Boyd-Graber, Blei, and Zhu (2007) は WordNet の語義を用いることでトピックモデルを教師なし WSD に応用した．Guo and Diab (2011) も同様にトピックモデルと WordNet の組み合わせの手法を提案している

が，概念構造は利用せず，辞書の定義文から事前学習を行う手法で，all-words WSD に関して Boyd-Graber らと同程度の精度を上げている．また，谷垣，撫中，匂坂 (2016) は階層ベイズとギブスサンプリングを用いた英語の all-words の WSD を提案している．

日本語の all-words WSD の研究には Baldwin, Kim, Bond, Fujita, Martinez, and Tanaka (2008), Komiya, Sasaki, Morita, Sasaki, Shinnou, and Kotani (2015) や Shinnou, Komiya, Sasaki, and Mori (2017) がある．日本語は表意文字（漢字）を利用しているため，すでに書かれた時点で意味が分かることが多い．そのため，日本語の WSD は英語の WSD に比べて，語義の差が小さいと考えられる．小さな語義の差を，あまり用例がない状態でも解かなければならない点が，日本語の all-words WSD の難しさであろう．Baldwin et al. (2008) では，machine readable dictionary (MRD) ベースの手法を提案している．Komiya et al. (2015) では，多義語の周辺に現れる語義の分布を利用する教師なし学習による周辺語義モデルを提案している．この論文の手法はギブスサンプリングを用いたシソーラスベースの all-words の WSD である．ただしこのシステムには EDR 電子化辞書による概念体系辞書が組み込まれており，再現するのが困難である．また，Shinnou et al. (2017) は単語分割をするテキスト分析のツールキットを応用し，教師ありの手法で all-words WSD を簡易に行えるシステムを作成している．

3 比較対象となるベースライン手法

本研究では，四つの比較対象となるベースライン手法を用いた．一つは語義をランダムに選択した場合の正解率 (random) である．コーパス中の全多義語の出現ごとの平均語義数の逆数により求めた．

二つ目は最頻出の語義をテキストコーパスから疑似的に推定する手法（Pseudo Most Frequent Sense. 以下 PMFS）である．この手法では，テストコーパスと同分野についての学習用テキストコーパスをテストコーパスとは別に用意し，語義の頻度を計算する．例えば，学習用コーパスに語義 a を持つ単語が出現した場合は語義 a の頻度に 1 を割り振る．また，語義 a と語義 b を持つ単語（多義語）が出現した場合は語義 a に 1/2，語義 b に 1/2 の頻度を割り振る．このようにして学習用コーパスでのすべての単語の語義の頻度を加算して，語義ごとに頻度を求めておく．テストの際には，WSD の対象単語のそれぞれの語義候補のうち，求めておいた頻度が最も高い語義を選択する．

三つ目は分類語彙表の分類番号を語義とした Yarowsky の手法 (Yarowsky 1992) である．Yarowsky の手法では，学習用コーパスから分類番号ごとに用例を集め，その中に出現する特徴の重みを計算しておく．ここでの特徴とは用例に出現する自立語である，語義 c における特徴 f の重み

は以下の式で定義される．

$$w(c,f) = \frac{\log Pr(f|c)}{Pr(f)} \tag{1}$$

対象単語の周辺の自立語の重みの合計を語義候補ごとに計算し，最も大きい値になった語義を選んでいく．

四つ目は，Boyd-Graber et al. (2007) で提案された，latent Dirichlet allocation with WordNet（以下 LDAWN）と呼ばれる手法である．LDAWN は，トピックモデル LDA (Latent Dirichlet Allocation) において，各トピックが持つ単語の確率分布を，概念辞書上の単語生成過程である WORDNET-WALK に置き換え，WSD に応用したモデルである．ルート概念からの経路の違いにより，語義の違いを表現している．WORDNET-WALK とは，WordNet や分類語彙表のような木構造の概念辞書において，ルート概念から下位概念への遷移を確率的に繰り返し，リーフ概念が表す単語を出力する単語生成過程である．LDAWN では，各文書が持つトピックの確率分布と，各トピックにおける各概念から下位概念への遷移確率分布をギブスサンプリングから求めている．WSD は対象単語のトピックを推定し，対応する遷移確率分布を用いてルート概念から対象単語までの経路を推定することで行える．

4　概念辞書の類義語と分散表現を利用した all-words WSD

単語の語義は周辺の文脈によって推定できることから，周辺の単語同士が類似している場合，中心の単語同士も類似していると考えられる．我々はこの考えをもとにした教師なしの all-words WSD の手法を提案する．

我々の提案する手法では，類義語を用いて語義を決定する．ここで，「犬」という単語の例を考える．「犬」という単語には「動物の犬」と「スパイ」という意味の二つの意味がある．どちらの意味なのか決定するために，我々は類義語の文脈と，対象単語の文脈を比較する．文脈を比較するためには，後述する「周辺単語ベクトル」を用いる．ある用例の周辺単語ベクトルが，「動物の犬」という意味を持つ類義語の周辺単語ベクトルよりも，「スパイ」という意味の周辺単語ベクトルに近ければ，その用例の語義は「スパイ」の方であると判定する．周辺単語ベクトルの作成方法として，我々は単語の分散表現を用いる方法，分類番号の分散表現を用いる方法，その両方を用いる方法の三種類の手法を提案する．

我々の手法は，WSD を繰り返して行う．はじめに，単語の分散表現を用いる方法で周辺単語ベクトルを求め，類義語の情報から文書内のすべての多義語の語義を推定する．単語の分散表現は語義タグのないテキストコーパスから求められるため，こうして教師なしの all-words WSD が実現できる．本研究の語義は分類語彙表の分類番号であるから，語義を推定した時点ですべての単語の分類番号を推定できる．その推定した分類番号をもとに，今度は分類番号の分散表

現を作成し，単語の分散表現を用いる方法と同様に，周辺単語ベクトルを求め，類義語の情報から文書内のすべての多義語の語義を推定することで，より正確な語義を推定することができる．この時点で推定された語義（分類番号）もまた，新たな分類番号の分散表現を作成するのに利用できる．我々はこれらの処理を繰り返すことで，最終的な語義を推定した．

4.1　概念辞書の類義語

本研究では，概念辞書として分類語彙表を使用した．分類語彙表とは単語を意味によって分類したシソーラスである．レコード総数は約 10 万件で，各レコードには「レコード ID 番号／見出し番号／レコード種別／類／部門／中項目／分類項目／分類番号／段落番号／小段落番号／語番号／見出し／見出し本体／読み／逆読み」という項目がある．分類番号は類・部門・中項目・分類項目を表す番号で，分類語彙表では主にこの番号によって単語が分類されている．その他にも “段落番号”，“小段落番号”，“語番号” が各レコードには割り振られており，それらすべての番号によってレコードが一意に決まるようになっている．さらに，分類語彙表には「意味的区切り」が 240 箇所に存在し，分類番号による分類をさらに細かく分けている．したがって分類の細かさは，分類番号による分類＜分類番号+意味的区切りによる分類＜分類番号+段落番号による分類＜分類番号+段落番号+小段落番号による分類（右に行くほど細かい）といえる．本研究ではこの分類語彙表から対象単語の類義語を求め，語義の予測に用いる．具体的には，“分類番号+意味的区切り” によって同じグループに分類された単語を類義語とする場合と “分類番号+段落番号” によって同じグループに分類された単語を類義語とする場合の 2 パターンで実験を行った．例えば「犬」という単語は分類語彙表中で二か所に存在する．すなわち，「犬」は二つの分類番号 (1.2420, 1.5501) を持つ多義語である．分類語彙表における「犬」の一部を表 1 に示す．

上記の条件で「犬」の類義語を求めると，分類番号：1.2410+意味的区切りで得られる類義語は［スパイ，回し者，…，教師，魔法使い，…］の 429 単語，1.5501+意味的区切りで得られる類義語は［おおかみ，くま，…，象，馬，…］の 302 単語となる．また，分類番号+段落番号：1.2410+27 で得られる類義語は［スパイ，回し者，…］の 11 単語，1.5501+04 で得られる類義語は［おおかみ，くま，…］の 74 単語となる．

表 1　分類語彙表における「犬」

類	部門	中項目	分類項目	分類番号	段落番号	小段落番号	語番号
体	主体	成員	専門的・技術的職業	1.2410	27	02	04
体	自然	動物	哺乳類	1.5501	04	01	01

4.2　単語の分散表現を用いる手法

周辺の単語同士が類似している場合，中心の単語同士の語義も類似している，と考え，本稿では以下の手法を提案する．まず，対象単語の周辺の単語（前後 2 単語ずつ）のそれぞれの分散表現（以下，w2v）を求める．そして，これらの w2v を連結し一つの分散表現にしたものを対象単語の「周辺単語ベクトル」とする．このとき，「.」や「(」などの記号は周辺単語に含めなかった．また，前後の単語数が 2 に満たないときは null とし，すべてゼロベクトルで補った．なお，本手法では，自立語以外についても上記の条件を満たせば，分散表現を作成した．

次に，分類語彙表から対象単語の語義候補ごとに類義語を求め，コーパス中に出現する類義語からも対象単語と同様に周辺単語ベクトルを作成していく．この際，類義語の周辺単語ベクトルには語義候補の語義をラベル付けしておく．また，4.1 節のようにして分類語彙表から類義語を求めた際に語義候補間で重複する類義語があればその単語はどちらからも除外した [2]．最後に，対象単語の周辺単語ベクトルとラベル付けした類義語の周辺単語ベクトル群との距離を計算し，K 近傍法によってラベルを一つ求めこれを対象単語の語義とした．なお，K 近傍法を利用したのはデータスパースネスに対して頑健であり，all-words WSD では処理対象となる，コーパス中に用例の少ない単語に対しても WSD を行うことが可能だからである．

例えば以下の文における「犬」の語義を提案手法を用いて予測してみる．

彼は警察の犬だ.

この文における「犬」の周辺単語は「警察」「の」「だ」「null」となり，単語 w の分散表現を $\mathrm{w2v_w}$ とすると周辺単語ベクトルは

$$\left[\mathrm{w2v_{警察}} \quad \mathrm{w2v_{の}} \quad \mathrm{w2v_{だ}} \quad \mathrm{w2v_{null}}\right]$$

となる．ここで null は文脈が文の外側に出る場合に素性が未定義であることを表す．次に「犬」の類義語を求め，用例を集める．「犬」の語義候補は分類番号 1.2410 と 1.5501 であり，4.1 節で述べたようにして類義語を求めると 1.2410 の類義語は［スパイ，回し者，…］，1.5501 の類義語は［きつね，くま，…］となるが，この中から多義語はすべて除外し（「くま」は多義語なので除外する）単義語のみ使用する．さらに，1.2410 の類義語と 1.5501 の類義語に重複する単語がある場合はどちらからも除外する．コーパス中に出現するこれらの類義語から，「犬」と同じようにして周辺単語ベクトルを作っていき（図 1，図 2），1.2410 または 1.5501 のラベルを付与する．最後に，「犬」の周辺単語ベクトルとラベルが付与された類義語の周辺単語ベクトルの距離を計算し，K 近傍法で「犬」の周辺単語と距離が近いラベルを求め語義を決定する．

[2] 予備実験によれば有効な手法であったため，このようにした．

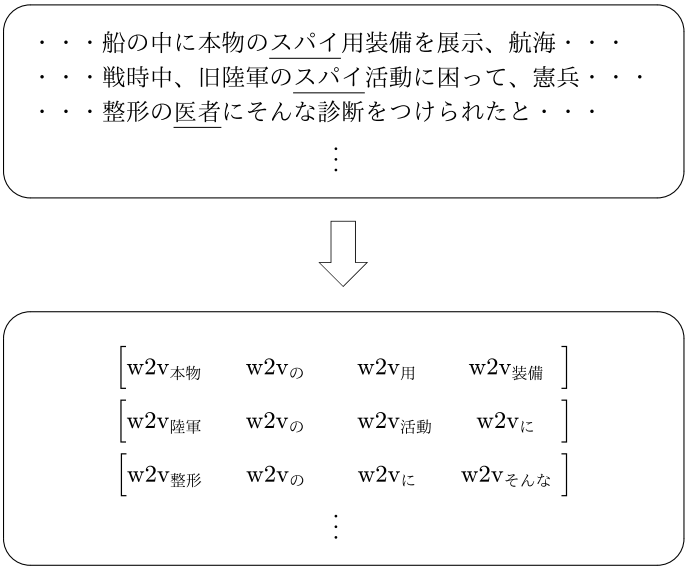

図 1　1.2410 の類義語の用例とその周辺単語ベクトル

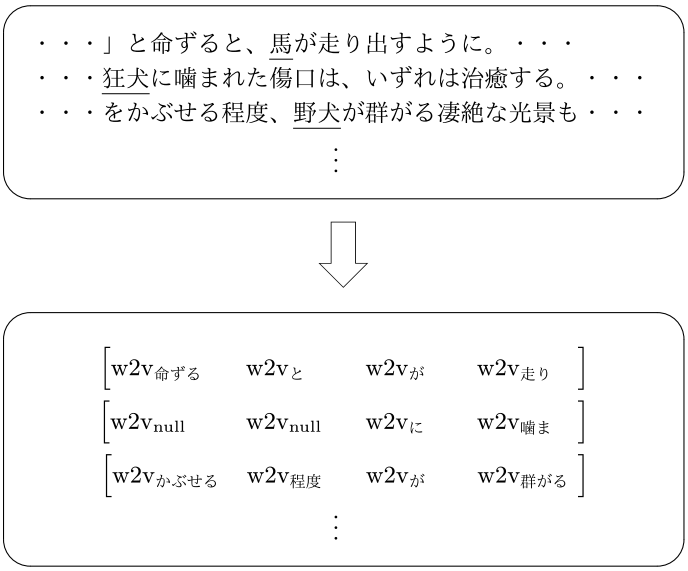

図 2　1.5501 の類義語の用例とその周辺単語ベクトル

4.3　分類番号の分散表現を用いる手法

本手法では，分類番号の分散表現（以下，c2v）を作成し語義の予測に用いる．まず，4.2 節の手法によって予測した結果をもとに，コーパス全体を分類番号の系列（語義列）に変換する．例えばコーパスが図 3 のような場合，“ビデオ” (1.4620) などは語義を 1 つしか持たない単義語で

あるためその語義に置き換え，“ジョッキー” (1.2410 or 1.2450) や“年” (1.1630 or 1.1962) などの多義語は 4.2 節の手法で予測した語義に置き換える．なお，“根本”や“が”や“「”のような分類番号を持たない単語は置き換えずにそのままとする．したがって，分類番号の系列は図 4 のようになる．こうして作成した分類番号の系列を分散表現作成ツール（gensim の Word2Vec[3]）に入力することで，テキストコーパスの単語列から w2v を作成したように，分類番号の系列から c2v を作成する．

次に，4.2 節の手法と同じようにして語義を予測していく．このとき，周辺単語ベクトルは w2v と c2v を組み合わせた場合と，c2v のみの場合の 2 通りとした．また，4.2 節の手法では類義語の中から多義語をすべて排除したが，そうすることで用例が少なくなるという問題点がある．そこで本手法では周辺単語ベクトルを作る際に，4.2 節の手法で語義候補の語義と予測された多義語の用例も追加した．前述の例文の「犬」の語義を予測するまでの流れを以下に示す．まずは「犬」の周辺単語ベクトルを作成する．周辺単語が「警察」,「の」,「だ」,「null」のとき，w2v+c2v で作成した周辺単語ベクトルは

$$\left[\mathrm{w2v}_{\text{警察}} \quad \mathrm{c2v}_{\text{警察}} \quad \mathrm{w2v}_{\text{の}} \quad \mathrm{c2v}_{\text{の}} \quad \mathrm{w2v}_{\text{だ}} \quad \mathrm{c2v}_{\text{だ}} \quad \mathrm{w2v}_{\text{null}} \quad \mathrm{c2v}_{\text{null}}\right]$$

となり，c2v で作成した周辺単語ベクトルは

根本要がビデオジョッキーを務め、1960 年代後半から 80 年代前半のロックを届ける．
自由党の「政策新人類」渡辺喜美衆院議員は提案する「経済非常事態」を宣言し産業再生委の創設を！
⋮

図 3　コーパスの例

根本 要 が 1.4620 1.2410 を 2.3320 、 1.1960 1.1960 1.1960 1.1960 1.1630 1.1623 1.1650 から 1.1960 1.1960 1.1630 1.16323 1.1650 の ロック を 2.1521 ．
1.2760 党 の 「 1.3084 3.1660 1.5501 」 渡辺 喜美 1.2730 1.2400 は 1.1210 2.1211 「 1.3710 1.1331 1.1000 」 を 1.3100 2.1211 1.3801 1.1211 委 の 1.1220 を ！
⋮

図 4　分類番号の系列

[3] https://radimrehurek.com/gensim/models/word2vec.html

$$\left[\mathrm{c2v}_{\text{警察}} \quad \mathrm{c2v}_{\text{の}} \quad \mathrm{c2v}_{\text{だ}} \quad \mathrm{c2v}_{\text{null}}\right]$$

となる．次にコーパス中に出現する類義語から周辺単語ベクトルを「犬」と同じようにして作成する．このとき，「犬」の 1.5501 における類義語には「くま」が含まれるが，「くま」は 2 つの分類番号 (1.1700, 1.5501) を持つ多義語である．この場合，4.2 節の手法によって 1.5501 と予測された「くま」の用例からも周辺単語ベクトルを作成する．最後に，4.2 節と同様に K 近傍法によって語義を求める．

5　評価実験

実験には BCCWJ に分類語彙表の分類番号がアノテーションされたコーパスを使用する．コーパス中のすべての多義語の語義を予測する問題設定とする．また，ここでいう多義語とは，複数の分類番号を持つ単語である．表 2，表 3 に使用したコーパスの文書数と統計を示す．なお，表 3 および以降における「トークン」の数とは出現したのべ数であり，「タイプ」とは種類の数を指す．特に，単語のタイプ数は語彙数に相当する．多義語（トークン）の平均語義数は，コーパス中の多義語の用例を無作為にひとつ選んだ際，その多義語が平均どのくらいの用例をコーパス中に持っているかを示している．表 3 において平均語義数は 2.98 なので，当て推量してみると 1/2.98(=0.336) の確率で正解になることを示している．

また，比較手法の PMFS と Yarowsky の手法で用いる学習用コーパスには BCCWJ の分類番号が付与されていない部分も含めて使用した．表 4，表 5 に使用した学習用コーパスの文書数と統計を示す．

提案手法と LDAWN では，テストコーパスのみを用いて実験を行った．

Yarowsky の手法において，学習用コーパスから用例を集める際は前後 10 単語ずつとした．

表 2　コーパスに含まれるジャンルとその文書数

書籍 (PB)	43
雑誌 (PM)	40
新聞 (PN)	110

表 3　コーパスの統計

単語数（トークン）	347,094
単語数（タイプ）	19,433
多義語数（トークン）	73,763
多義語数（タイプ）	4,190
多義語（トークン）の平均語義数	2.98

表 4　学習用コーパスに含まれるジャンルとその文書数

書籍 (PB)	533
雑誌 (PM)	496
新聞 (PN)	1,363

表 5　学習用コーパスの統計

単語数（トークン）	4,688,762
単語数（タイプ）	63,286
多義語数（トークン）	1,012,357
多義語数（タイプ）	7,137
多義語（トークン）の平均語義数	2.99

さらに，対象単語の周辺単語の重みの和を求めるときも，前後 10 単語ずつから求めた．

LDAWN では，Komiya et al. (2015) を参考にパラメータを設定した．具体的には 1 文を 1 文書とし，メタパラメータである K （トピック数）は 32，S （遷移確率の調整定数）は 10 そして τ （ディリクレ分布の prior）は 0.01 とした．

提案手法では類義語を分類語彙表から“分類番号+意味的区切り”を用いて求める場合と“分類番号+段落番号”を用いて求める場合の 2 通りで実験を行った．w2v の作成には nwjc2vec (新納，浅原，古宮，佐々木 2017) を使用した．nwjc2vec とは，国語研日本語ウェブコーパス (NWJC) に対して word2vec (Mikolov, tau Yih, and Zweig 2013c; Mikolov, Chen, Corrado, and Dean 2013a; Mikolov, Sutskever, Chen, Corrado, and Dean 2013b) で学習を行った分散表現データである．word2vec のパラメータは，アルゴリズムに Continuous Bag-of-Words (C-BoW) を利用し，次元数を 200，ウィンドウ幅を 8，ネガティブサンプリングに使用する単語数を 25，反復回数を 15 としている．c2v は，コーパスを分類番号の系列に変換したものを同じく word2vec で学習して作成した．その際，アルゴリズムは C-BoW を利用し，次元数を 50，ウィンドウ幅を 5，ネガティブサンプリングに使用する単語数を 5，反復回数を 3，min-count を 1，として学習を行った．また，周辺単語ベクトルを作成する際，周辺に単語が四つない場合（対象単語が文頭や文末にある場合など null に相当する）や，word2vec で学習されていない単語の分散表現などは，同じ次元の零行列を用いた．周辺単語ベクトルの作成に用いる周辺単語の数は前後 2 単語ずつの 4 単語とした．したがって，w2v のみで作成した周辺単語ベクトルは 800 次元，w2v+c2v で作成した周辺単語ベクトルは 1000 次元となる．周辺単語ベクトルの距離を測り K 近傍法で分類する過程には scikit-learn[4]の KNeighborsClassifier を使用した．ここではユークリッド距離を使用し，k=1，3，5，weight=uniform，distance（uniform=重みなし，distance=重みあり）で

[4] https://scikit-learn.org/stable/

実験を行った．

実験では，最も良いパラメータを選ぶため，w2v を利用した場合，w2v+c2v を利用した場合，c2v を利用した場合のそれぞれに対し，パラメータ三種類（“分類番号+意味的区切り”/“分類番号+段落番号”，k=1/3/5，uniform/distance）のバリエーションを試した．この際，w2v を利用した場合では一度目（繰り返しなし），w2v+c2v を利用した場合と c2v を利用した場合では二度目（一度目は w2v を利用し，二度目で w2v+c2v または c2v を利用して繰り返した）の結果で比較する．また，この際に最も良かった設定について五度繰り返して正解率を見た．

6 結果

四つの比較手法および三つの提案手法の結果を表 6 に示す．提案手法の結果は，w2v を利用した手法，w2v+c2v を利用した手法，c2v を利用した手法における，それぞれ最も結果が良かった場合のパラメータを利用した際の結果である．パラメータについては考察で述べる．

また，表 7 に最良の場合の手法とパラメータを利用した場合の繰り返しによる正解率の変化を示す．最も良い数値を太字で示した．

表 6 から，提案手法 w2v+c2v と提案手法 c2v が，比較手法である random，PMFS，Yarowsky の手法，LDAWN のすべてを上回る結果となったことが分かる．なお，比較手法の中では PMFS の正解率が最も高い結果となった．w2v を利用した手法，w2v+c2v を利用した手法，c2v を利用した手法の三手法を比較すると，w2v+c2v を利用した手法が最もよく，続いて c2v を利用した手法となり，このことから c2v の利用が all-words WSD において有効であることが示された．

繰り返しの効果について表 7 を見てみると，w2v だけを用いた一度目の結果よりも，c2v も併

表 6 比較手法と提案手法の正解率 (%)

random	33.6
PMFS	52.1
Yarowsky	44.0
LDAWN	39.5
提案手法 w2v	51.3
提案手法 w2v+c2v	**54.1**
提案手法 c2v	53.5

表 7 繰り返しによる正解率の変化 (%)

繰り返し回数	0	1	2	3	4	5
分散表現	w2v	w2v+c2v				
正解率	51.3	54.1	**54.2**	**54.2**	**54.2**	**54.2**

せて用いた二度目の結果（一度繰り返したとき）の方が正解率が上昇している[5]．これに対して三度目の結果は僅かに上昇し，その後は変化がない．このため，c2v を導入した繰り返しに効果はあるが，何度も繰り返しても正解率はあまり変わらないことが分かった．

7 考察

7.1 提案手法のパラメータ

提案手法のパラメータ別の結果を表 8 に示す．一つの実験の中で最も良い数値を太字とした．また，提案手法の中で最も良い数値には下線を引いた．さらに，比較手法すべてよりも良い結果となったものを斜体で示した．

表 8 から，提案手法で最も良い結果となったのは“分類番号+意味的区切り”，w2v+c2v を用いた場合であることが分かる．また，提案手法では K の値や重みの有無によって精度に大きな差がないことが分かった．類義語の決め方に注目すると，“分類番号+意味的区切り”を使用するほうが“分類番号＋段落番号”を利用した場合に比べて，常に良い結果となっていることがわかる．類義語の区分に“分類番号＋段落番号”を利用した場合には，c2v の導入によって正解率が下がっている．このことから，類義語の区分には適切なものを利用する必要があることが分かる．

比較手法の結果と提案手法の正解率を比べると，“分類番号+意味的区切り”，w2v+c2v または“分類番号+意味的区切り”，c2v を用いた場合（表の 4〜7 行目）では random，PMFS，Yarowsky

表 8 パラメータごとの正解率 (%)

分散表現	類義語の区分	重み	K=1	K=3	K=5
w2v	分類番号+意味的区切り	uniform	**51.3**	**51.3**	51.1
w2v	分類番号+意味的区切り	distance	**51.3**	**51.3**	51.1
w2v+c2v	分類番号+意味的区切り	uniform	*53.9*	*53.8*	*53.6*
w2v+c2v	分類番号+意味的区切り	distance	*53.9*	***54.1***	*53.8*
c2v	分類番号+意味的区切り	uniform	*52.8*	*53.0*	*53.4*
c2v	分類番号+意味的区切り	distance	*52.8*	*53.2*	**53.5**
w2v	分類番号+段落番号	uniform	**51.2**	50.1	48.8
w2v	分類番号+段落番号	distance	**51.2**	50.4	48.9
w2v+c2v	分類番号+段落番号	uniform	49.2	**49.3**	**49.3**
w2v+c2v	分類番号+段落番号	distance	49.2	**49.3**	**49.3**
c2v	分類番号+段落番号	uniform	49.7	**49.8**	49.6
c2v	分類番号+段落番号	distance	49.7	**49.8**	49.7

[5] 予備実験では，w2v だけを利用して繰り返すパターンも試したが，正解率はあまり上昇しなかった．

の手法，LDAWN のすべてを上回る結果となった．

7.2 他手法との比較

Yarowsky の手法と本研究の提案手法は，類義語の周辺の単語を用いるという点では同じである．Yarowsky の手法は計算量が提案手法よりも少ないことから，大きな学習用コーパスから用例を集めることや，周辺単語として前後 20 単語を利用することができる．一方，提案手法は計算量が多いため，学習用コーパスは用意せずに WSD の対象となるコーパスから用例を集め，周辺単語も前後 4 単語しか利用していない．それでも Yarowsky の手法を上回る結果となったことから，分散表現が WSD において有効であることがわかる．

また，提案手法は LDAWN も上回った．しかも LDAWN は Yarowsky の手法よりも劣った．これには二つの原因が考えられる．一つは本研究のタスクが all-words WSD であることである．教師なし WSD の研究は多いが，それらの手法がそのまま教師なし all-words WSD において高精度の結果を出せるわけではない．トピックモデルを利用した手法がそのような手法の一つだと考えられる．トピックモデルを利用した教師なし WSD は，本質的に，対象単語の文脈をトピック分布で表現し，その分布が語義ごとに異なることを利用する．しかし語義の違いを区別するためのトピックの分割が全ての単語で同一である保証はない．また LDAWN では妥当な分割を求める手がかりとして語義の階層構造を利用するが，その階層構造として，分類語彙表の語義の階層構造が適切であるかどうかも疑問である．一方，提案手法は単語や語義の分散表現を利用しており，WSD の対象単語に依存しない．このため，より all-words WSD に適した手法となっている．二つ目は WSD で利用する対象単語の文脈情報として，トピック分布だけでは不十分であることである．トピックは大域的な文脈情報である．しかし実際に WSD で有効な情報は，直前直後の出現単語といった局所的な文脈情報である，LDAWN はそのような局所的な情報を直接的には利用していない．一方，本手法や Yarowsky の手法は直接的に局所的な文脈情報を利用しているために，LDAWN を上回ったと考えられる．特に日本語は表意文字を利用しているため，英語に比べて語義同士の意味が近い．そのため，トピックのような大域的な文脈よりも，周辺の単語のような局所的な文脈が効いたものと考えられる．また，提案手法では分類語彙表に多く含まれる，類義語の情報を利用しているため，上位下位概念よりも多くの情報が利用できる．さらに提案手法は分散表現を利用しているため，同一の単語でなくても類似度が計算でき，K 近傍法を利用したことによってデータスパースネスに強い手法となっている．

7.3 類義語についての考察

提案手法では分類語彙表から類義語を求めて WSD に利用する．類義語は，対象単語と意味が近いほど類義語として好ましい．しかし，意味が近い単語に限定しすぎると類義語の数，すなわち類義語の用例の数が少なくなってしまい語義の予測に影響が出てしまう．例えば，「犬」

の分類番号 1.2410 の類義語になりうる単語を列挙すると表 9 のようになる．分類番号が等しく多義語でない単語を類義語とした場合（分類番号 1.2410 には意味的区切りは存在しない.）「犬」の類義語は 429 単語となり，その中には「教育家」や「教師」,「神官」などの単語も含まれる．しかしこれらの単語は「犬」の語義とそれほど近い語義を持っているわけではなく,「犬」の語義を予測する場合に役立っているとは考えにくい．一方,“分類番号＋段落番号”が等しい単語

表 9　「犬」の分類番号 1.2410 の類義語になりうる単語（一部省略）

分類番号	段落番号	小段落番号	語番号	見出し
⋮	⋮	⋮	⋮	⋮
1.2410	03	01	01	教育家
1.2410	03	01	02	教育者
1.2410	03	02	01	教師
⋮	⋮	⋮	⋮	⋮
1.2410	03	06	05	インストラクター
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
1.2410	27	01	02	密偵
1.2410	27	01	03	スパイ
1.2410	27	01	04	諜報員
1.2410	27	02	01	間者
1.2410	27	02	02	間諜
1.2410	27	02	03	回し者
1.2410	27	03	01	二重スパイ
1.2410	27	03	02	ダブルスパイ
1.2410	27	04	02	工作員
1.2410	27	05	01	忍者
1.2410	27	05	02	忍びの者
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
1.2410	28	01	01	聖職者
1.2410	28	01	02	宗教家
1.2410	28	02	01	神官
⋮	⋮	⋮	⋮	⋮
1.2410	28	03	02	社司
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮

表 10　一つの語義から獲得した類義語の用例数の平均

	多義語全体	正解した多義語	不正解した多義語
分類番号+意味的区切り	368.0	571.8	213.2
分類番号+段落番号	163.3	329.8	36.9

を類義語とした場合は「犬」とかなり意味的に近い単語に限定され，これらの単語は「犬」と言い換えても文脈が変化しないため類義語としてふさわしい単語だといえる．しかしその数はわずか 11 単語となり類義語の用例の数が大きく減少してしまう．

そこで，本研究で用いたコーパスにおいて多義語が一つの語義から類義語の用例をいくつ獲得できるか平均を求めると，表 10 のようになった．正解・不正解は w2v を利用した場合（表 8 の 2，3，8，9 行目）の結果である．

多義語全体の平均を見ると，“分類番号＋意味的区切り”に比べて分類番号+段落番号を類義語の定義として用いた場合は獲得できる用例の数が大幅に少なくなっていることがわかる．本研究では，“分類番号＋段落番号”で類義語を集めた場合，“分類番号＋意味的区切り”で類義語を集めた場合と比べて精度が下がっているが，これは獲得できる類義語の用例数が著しく減少したことが原因だと考えられる．また，正解した多義語が獲得した用例数の平均が不正解した場合を大きく上回っていることからも，語義を正しく予測するには類義語の用例数をある程度多く獲得する必要があることが分かった．ただし，最も多く用例を獲得できた語義は“分類番号+意味的区切り”だと 2.1200 で用例の数は 5,535 個，“分類番号+段落番号”だと 1.1960-01 で用例の数は 3,974 個であり，用例の数が最も少なかったのは 2.1340 や 3.1522-05 で用例の数は 1 個だった．このことから，分類語彙表を用いて類義語の用例を獲得する場合用例の数は語義ごとにかなりばらつきが生まれることも確認できた．したがって本研究の提案手法は，用例が極端に少ない場合には学習用コーパスを用意して用例を獲得したり，広い意味で類義語を定義して用例の数を増やしたりすることで精度が向上する可能性が考えられる．また，用例が極端に多い場合は，段落番号や小段落番号を用いるなど，類義語を狭い意味で定義し，対象単語により意味が近い単語に限定することで精度が向上し，さらに計算量を減少させることができると考えられる．

8　おわりに

本稿では，教師なしによる日本語の all-words WSD の手法を提案した．具体的には，対象単語の周辺単語ベクトルと対象単語の類義語の周辺単語ベクトルを作成し，それらの距離を計算して K 近傍法によって語義を求める．周辺単語ベクトルは，前後 2 単語ずつの w2v を連結した

ベクトル，w2v，c2v を連結したベクトル，c2v を連結したベクトルの 3 通りで実験を行った．c2v は予測結果をもとにコーパスを分類番号の分かち書きに変換して作成した．類義語は分類語彙表から“分類番号+意味的区切り”，“分類番号+段落番号”の 2 通りの方法で定義し，それぞれで実験を行った．実験の結果，w2v+c2v，“分類番号+意味的区切り”を用いた場合が最も高い精度となった．また，提案手法ではランダムベースライン，PMFS，Yarowsky の手法，LDAWN を超える精度を出すことができ，語義曖昧性解消において有効な手法であることが確認できた．さらに結果を分析すると，不正解だった対象単語の 1 語義当たりの類義語の用例数が正解だった場合と比べて少ない傾向にあることや，獲得できる類義語の用例数は語義によってばらつきがあることが確認できた．これらのことから，本手法で精度をさらに向上させる方法として，類義語の用例を獲得しづらい語義では学習用コーパスから用例を獲得する，類義語の意味の幅を広くするなどの方法で用例数を確保することが考えられる．

謝　辞

本研究は，国立国語共同研究プロジェクト「コーパスアノテーションの拡張・統合・自動化に関する基礎研究」および「all-words WSD システムの構築および分類語彙表と岩波国語辞典の対応表作成への利用」の研究成果を含んでいます．また，本研究は JSPS 科研費 15K16046 および 18K11421 の助成と，茨城大学女性エンパワーメントプロジェクトの助成を受けたものです．また，本論文の内容の一部は，11th edition of the Language Resources and Evaluation Conference で発表したものです (Suzuki, Komiya, Asahara, Sasaki, and Shinnou 2018).

参考文献

Baldwin, T., Kim, S. N., Bond, F., Fujita, S., Martinez, D., and Tanaka, T. (2008). “MRD-based Word Sense Disambiguation: Further Extending Lesk.” In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008)*, pp. 775–780.

Boyd-Graber, J., Blei, D., and Zhu, X. (2007). “A Topic Model for Word Sense Disambiguation.” In *EMNLP-CoNLL-2007*, pp. 1024–1033.

Gale, W. A., Church, K. W., and Yarowsky, D. (1992). “One Sense Per Discourse.” In *Proceedings of the Workshop on Speech and Natural Language*, HLT ’91, pp. 233–237, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guo, W. and Diab, M. (2011). “Semantic Topic Models: Combining Word Distributional Statistics and Dictionary Definitions.” In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 552–561.

Kato, S., Asahara, M., and Yamazaki, M. (2018). "Annotation of 'Word List by Semantic Principles' Labels for Balanced Corpus of Contemporary Written Japanese." In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC 32)*.

小林健人，白井清昭 (2018). 分類語彙表の分類項目を識別する語義曖昧性解消―Yarowsky モデルの適応と拡張―. 言語処理学会第 24 回年次大会発表論文集, pp. 244–247.

Komiya, K., Sasaki, Y., Morita, H., Sasaki, M., Shinnou, H., and Kotani, Y. (2015). "Surrounding Word Sense Model for Japanese All-words Word Sense Disambiguation." In *PACLIC 2015*, pp. 35–43.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). "Efficient Estimation of Word Representations in Vector Space." In *Proceedings of ICLRWorkshop 2013*, pp. 1–12.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). "Distributed Representations of Words and Phrases and their Compositionality." In *Proceedings of NIPS 2013*, pp. 1–9.

Mikolov, T., tau Yih, W., and Zweig, G. (2013c). "Linguistic Regularities in Continuous Space Word Representations." In *Proceedings of NAACL 2013*, pp. 746–751.

Okumura, M., Shirai, K., Komiya, K., and Yokono, H. (2011). "On SemEval-2010 Japanese WSD Task." 自然言語処理, **18** (3), pp. 293–307.

新納浩幸，浅原正幸，古宮嘉那子，佐々木稔 (2017). nwjc2vec: 国語研日本語ウェブコーパスから構築した単語の分散表現データ. 自然言語処理, **24** (5), pp. 705–720.

Shinnou, H., Komiya, K., Sasaki, M., and Mori, S. (2017). "Japanese all-words WSD system using the Kyoto Text Analysis ToolKit." In *Proceedings of PACLIC 2017, no. 11*, pp. 392–399.

新納浩幸，佐々木稔，古宮嘉那子 (2015). 語義曖昧性解消におけるシソーラスの利用問題. 言語処理学会第 21 回年次大会発表論文集, pp. 59–62.

Sugawara, H., Takamura, H., Sasano, R., and Okumura, M. (2015). "Context Representation with Word Embeddings for WSD." In *Proceedings of PACLING 2015*, pp. 108–119.

Suzuki, R., Komiya, K., Asahara, M., Sasaki, M., and Shinnou, H. (2018). "All-words Word Sense Disambiguation Using Concept Embeddings." In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC 2018)*, pp. 1006–1011.

谷垣宏一，撫中達司，匂坂芳典 (2016). 語の出現と意味の対応の階層ベイズモデルによる教師なし語義曖昧性解消. 情報処理学会論文誌, **57** (8), pp. 1850–1860.

Yamaki, S., Shinnou, H., Komiya, K., and Sasaki, M. (2016). "Supervised Word Sense Disambiguation with Sentences Similarities from Context Word Embeddings." In *Proceedings of PACLIC 2016*, pp. 115–121.

Yarowsky, D. (1992). “Word-sense Disambiguation using Statistical Models of Roget’s Categories Trained on Large Corpora.” In *Proceedings of COLING*, pp. 454–460.

Yarowsky, D. (1995). “Unsupervised Word Sense Disambiguation Rivaling Supervised Methods.” In *ACL 1995*, pp. 189–196.

略歴

鈴木　類：2017 年茨城大学工学部情報工学科卒．2019 年茨城大学大学院理工学研究科情報工学専攻修了．

古宮嘉那子：2005 年東京農工大学工学部コミュニケーション工学科卒．2009 年同大学大学院博士後期課程電子情報工学専攻修了．博士（工学）．同年東京工業大学精密工学研究所研究員，2010 年東京農工大学工学研究院特任助教，2014 年茨城大学工学部情報工学科講師．現在に至る．自然言語処理の研究に従事．情報処理学会，人工知能学会，言語処理学会各会員．

浅原　正幸：2003 年奈良先端科学技術大学院大学情報科学研究博士後期課程修了．2004 年より同大学助教．2012 年より人間文化研究機構国立国語研究所コーパス開発センター特任准教授．2019 年より同教授．博士（工学）．言語処理学会，日本言語学会，日本語学会各会員．

佐々木　稔：1996 年徳島大学工学部知能情報工学科卒業．2001 年同大学大学院博士後期課程修了．博士（工学）．2001 年 12 月茨城大学工学部情報工学科助手．現在，茨城大学工学部情報工学科講師．機械学習や統計的手法による情報検索，自然言語処理等に関する研究に従事．言語処理学会，情報処理学会各会員．

新納　浩幸：1985 年東京工業大学理学部情報科学科卒業．1987 年同大学大学院理工学研究科情報科学専攻修士課程修了．同年富士ゼロックス，翌年松下電器を経て，1993 年より茨城大学工学部．現在，茨城大学工学部情報工学科教授．博士（工学）．機械学習や統計的手法による自然言語処理の研究に従事．言語処理学会，情報処理学会，人工知能学会各会員．

（2018 年 11 月 1 日　受付）
（2019 年 2 月 8 日　再受付）
（2019 年 3 月 18 日　採録）