

## 依存構造に基づく単語から語義の分散表現への細分化

芦原, 和樹  
大阪大学大学院情報科学研究科

梶原, 智之  
大阪大学データビリティフロンティア機構

荒瀬, 由紀  
大阪大学大学院情報科学研究科

内田, 諭  
九州大学大学院言語文化研究院

<https://hdl.handle.net/2324/2244113>

---

出版情報：情報処理学会研究報告．自然言語処理研究会報告．2018-NL-237 (3), pp.1-7, 2018-09-18.  
Information Processing Society of Japan (IPSJ)

バージョン：

権利関係：利用は著作権の範囲内に限られる

# 依存構造に基づく単語から語義の分散表現への細分化

芦原 和樹<sup>1,a)</sup> 梶原 智之<sup>2,b)</sup> 荒瀬 由紀<sup>1,c)</sup> 内田 諭<sup>3,d)</sup>

**概要:** 多くの自然言語処理タスクにおいて単語分散表現が利用されている。しかし、各単語に1つの分散表現を割り当てるアプローチでは、多義語における各語義の情報が混在してしまう。この問題に対処するために、先行研究では品詞やトピックごとに異なる分散表現を割り当てたが、これらの手法には多義性を扱う粒度が粗いという課題がある。本研究では、単語間の依存関係を手がかりとして各単語に複数の分散表現を割り当てる手法を提案する。提案手法は、先行研究よりも細かい粒度で多義性を扱うことができる反面、データスパースネス問題が危惧される。そこで我々は、多義語における各分散表現の初期値として、語義を区別せずに事前学習した分散表現を用いることでこの問題に対処する。単語間の意味的類似度推定タスクおよび語彙的換言タスクにおける実験の結果、提案手法は各単語に複数の分散表現を割り当てる先行研究よりも高い性能を発揮した。また、学習データ中の出現回数が与える影響についての分析の結果、事前学習した分散表現を用いることがデータスパースネス問題を解決するために有効であることも確認できた。

## 1. はじめに

分布仮説 [1] に基づき単語の意味的あるいは構文的な情報を密ベクトルで表現する単語分散表現 [2–6] が、機械翻訳 [7]、文書分類 [8] および語彙的換言 [9] など多くの自然言語処理応用タスクにおける性能改善に大きく貢献してきた。単語分散表現は今やこれら応用タスクの基盤となっており、その性能改善は重要な課題である。広く利用されている CBOW (Continuous Bag-of-Words) [2] や SGNS (Skip-gram with Negative Sampling) [3] などのアルゴリズムでは、各単語に1つの分散表現を割り当てるため、多義語の分散表現に各語義の情報が混在するという問題がある。

文脈に応じて分散表現を使い分けるために、多義語に複数の分散表現を割り当てる手法 [10–13] が提案されている。しかし、語義曖昧性解消はそれ自身が難しいタスクであるため、これらの先行研究では近似的なアプローチを用いている。例えば、Paetzold and Specia [11] は品詞ごとに、Fadae ら [12] はトピックごとに異なる分散表現を割り当てるが、これらの手法には多義性を扱う粒度が粗いという課題がある。以下の例では、いずれの文もトピックは food であり、単語 **soft** の品詞は形容詞である。

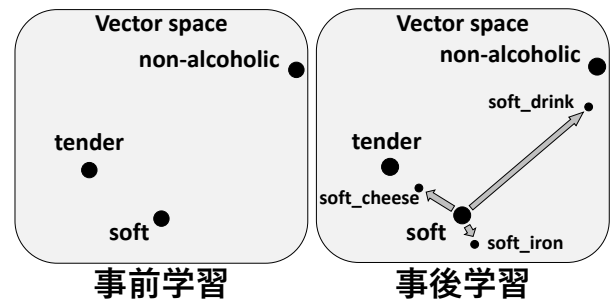


図 1 提案手法の概要: 各単語を依存関係のある語とペアにして分散表現を学習する。データスパースネス問題に対処するため、単語単位で学習した分散表現による初期化を行う。

ex. 1) *I ate a **soft** cheese.*

ex. 2) *I drunk **soft** drinks.*

先行研究ではこれらの単語 **soft** を同じ分散表現で表す。しかし、ex. 1) の単語 **soft** は **tender** という意味を、ex. 2) の **soft** は **non-alcoholic** という意味を表す。このような品詞やトピックでは区別できない多義性を考慮するために、本研究では各単語により細かい粒度で複数の分散表現を割り当てる手法を提案する。

本研究の手法では語義を区別するために、各単語と依存関係にある単語 (context-word) を組み合わせる。各単語には、この対ごとに分散表現を生成する。例えば、ex. 1) の単語 **soft** は **soft\_cheese** という分散表現を、ex. 2) の **soft** は **soft\_drink** という分散表現を獲得する。これによって、提案手法では品詞やトピックよりも細かい粒度で語義を区別することができる。しかし、context-word の増

<sup>1</sup> 大阪大学大学院情報科学研究科  
<sup>2</sup> 大阪大学データリテリフロンティア機構  
<sup>3</sup> 九州大学大学院言語文化研究院  
a) ashihara.kazuki@ist.osaka-u.ac.jp  
b) kajiwara@ids.osaka-u.ac.jp  
c) arase@ist.osaka-u.ac.jp  
d) uchida@fkc.kyushu-u.ac.jp

加に伴い各単語が多くの分散表現を持つことになるため、我々の提案手法ではデータスパースネス問題に対処する必要がある。そこで各単語を原形化し、語義を区別せずに事前学習した分散表現によって多義語の各分散表現を初期化する(図1)。これによって、低頻度な context-word が与える分散表現は初期値の近傍に留まり、学習回数の少ない状況でも大きく意味が外れないことが期待できる。

提案手法の性能を評価するため、多義性を考慮する分散表現が特に重要な単語間の意味的類似度推定タスク [14] および語彙的換言タスク [15, 16] で実験を行った。評価の結果、提案手法は特に語彙的換言タスクにおいて、各単語に複数の分散表現を割り当てる先行研究 [10–12] よりも顕著に高い性能を発揮した。また、詳細な分析の結果、初期化がデータスパースネス問題を解決する上で有効であることも確認できた。

## 2. 関連研究

Li and Jurafsky [17] は、各単語に複数の分散表現を与えることが、品詞付与や文の意味的類似度推定など多くの自然言語処理タスクの性能改善に貢献することを示した。

各単語に複数の分散表現を割り当てる手法として、以下に示す先行研究 [10–13] がある。Athiwaratkun and Wilson [13] は、全ての単語が定数個の語義を持つと仮定し、各単語に複数の分散表現を割り当てた。彼らは語義数として2または3を仮定しているが、語義数は単語ごとに変化する。Neelakantan ら [10] は文脈の類似度に基づいて語義に相当するクラスタリングを行い、各単語に対してクラスタごとに分散表現を割り当てた。しかし、語義曖昧性解消はそれ自身が難しいタスクであり、クラスタリングの性能が分散表現学習の性能に影響を与えてしまう。Paetzold and Specia [11] は、各単語に対して品詞ごとに分散表現を割り当てた。品詞付与は解析誤りの影響が小さいという利点を持つが、1章で示した ex. 1) と ex. 2) のように同じ品詞の中でも更に多義性を持つ単語が存在する。Fadaee ら [12] は、各単語に対してトピックごとに分散表現を割り当てた。トピックは品詞よりも細かい粒度で語義を区別できるが、ex. 1) と ex. 2) のように更に細かい粒度で語義を区別することが望まれる場合がある。

## 3. 提案手法

本研究では、各単語に対してより細かい粒度で複数の分散表現を生成する。本研究の手法では、語義を区別するための手がかりとして、ある文において依存関係にある単語 (context-word) に注目する。本研究では、意味表現においてより有効と考えられる、内容語 (名詞、動詞、形容詞、副詞) に限定して context-word として用いる。各単語  $w_i$  は context-word の集合  $C$  の要素ごとに異なる意味を持つと仮定し、 $w_i$  に対して  $|C|$  だけ分散表現を生成する。こ

### Algorithm 1 context-word を用いた分散表現の生成

```
Input: Set of training sentences  $S$ , Window size  $W$  in CBOW( $\cdot$ )  
for all  $s \in S$  do  
   $D \leftarrow \text{DEPPARSE}(s)$  ▷ 依存構造解析  
  for  $i \leftarrow 1, n$  do ▷  $n$ :  $s$  の 1 文長  
     $w_i \leftarrow s[i]$   
     $l_c \leftarrow s[i - W : i]$  ▷  $w_i$  の左側  $W$  単語  
     $r_c \leftarrow s[i : i + W]$  ▷  $w_i$  の右側  $W$  単語  
     $C \leftarrow \text{GETCONTEXTW}(w_i, D)$   
    for  $c \in C$  do  
       $w_c \leftarrow w_i + \text{"\_"} + c$  ▷ context-word と連結  
       $\text{CBOW}(w_c, l_c, r_c)$   
    end for  
  end for  
function  $\text{GETCONTEXTW}(w, D)$   
   $C \leftarrow \emptyset$   
  for all  $h, d \in D$  do ▷  $h$ : 係り元,  $d$ : 係り先  
    if  $h = w$  then  
       $C \leftarrow C + d$   
    else if  $d = w$  then  
       $C \leftarrow C + h$   
    end if  
  end for  
  return  $C$   
end function
```

で、context-word の数  $|C|$  は語義数よりも多くなる。例えば、soft\_cheese と soft\_candy が同じ tender という意味を表す分散表現として生成される。本研究では各分散表現は互いに干渉しないと仮定し、独立して学習、使用するものとする。

Algorithm 1 に提案手法の疑似コードを示す。まず、学習データ  $S$  に含まれる各文  $s$  について、依存構造解析によって依存関係  $D$  を得る。次に、各単語  $w_i \in s$  について  $\text{GETCONTEXTW}(w_i, D)$  によって context-word の集合  $C$  を獲得する。そして、単語  $w_i$  と context-word  $c \in C$  を連結し、 $w_c$  とする。最後に、 $W$  単語の左文脈  $l_c$  および右文脈  $r_c$  とともに  $w_c$  を  $\text{CBOW}(w_c, l_c, r_c)$  へ入力し、CBOW のアルゴリズムを用いて語義の分散表現を訓練する。ここで、 $W$  は CBOW における窓幅である。

図1の例では、soft の context-word として cheese, drink および iron が存在する。提案手法では soft\_cheese, soft\_drink および soft\_iron のように context-word ごとに分散表現を学習する。

我々の提案手法では、単語とその context-word の対ごとに分散表現を学習するため、各対の出現頻度が小さいというデータスパースネス問題に対処する必要がある。具体的には以下の2つの対策を取る。

対策1) 形態素解析により各単語を原形化

対策2) 事前学習によって分散表現を初期化

対策1) では、原形化によって語彙数を削減し、データスパースネス問題に対処する。対策2) では、context-word を考

慮しない通常の CBOW の学習を通して単語 `soft` の分散表現を得る (事前学習)。これを初期値として, `context-word` を考慮する語義ごとの分散表現を学習する (事後学習)。

共起頻度の低い単語と `context-word` の対は上手く学習が進まないと考えられるが, 対策 2) によってこのような分散表現は初期値の近傍に留まり, 大きく意味が外れないと期待できる。図 1 の例は, `soft_cheese`, `soft_drink` および `soft_iron` の分散表現が, それぞれ単語 `soft` の分散表現を起点に変化していく様子を示している。事後学習の結果, 単語 `soft` の主要な意味を持つ `soft_cheese` は初期値 `soft` の近くに位置し, 異なる意味を持つ `soft_drink` は `non-alcoholic` に近づく。なお, 共起頻度の低い `soft_iron` は, 初期値 `soft` の近傍に留まる。

## 4. 実験設定

提案手法の有用性を検証するために, 単語間の意味的類似度推定タスク [14] および語彙的換言タスク [15, 16] において実験を行う。両タスクとも所与の文脈における語義を考慮する必要があり, 多義性を考慮した分散表現が重要となる。

### 4.1 前処理

分散表現の学習のために, English Wikipedia<sup>\*1</sup>の本文約 10 億文を利用する。これらの文は, Stanford Parser [18] によって形態素解析され, 各単語が原形化される。ここで, 本実験では出現頻度 200 以下の単語を `<unk>` タグに置換し, 112,087 語のみを利用する。さらに, 同じく Stanford Parser を用いて依存構造解析を行い, 各単語の `context-word` を特定する。事前学習および事後学習においては, CBOW [2] のアルゴリズムを用いてエポック数 20, 窓幅 5 で 300 次元の分散表現を得る。

### 4.2 比較手法

本実験では, 各単語に対して 1 つあるいは複数の分散表現を生成する以下の手法を, 提案手法と比較する。

#### CBOW [2]

CBOW のアルゴリズムを用いて各単語に 1 つの分散表現を割り当てる。

#### SGNS [3]

SGNS のアルゴリズムを用いて各単語に 1 つの分散表現を割り当てる。

#### MSSG [10]

文脈の類似度に基づくクラスタリングによって各単語に複数の分散表現を割り当てる。

#### POS [11]

各単語に品詞ごとの分散表現を割り当てる。

表 1 SCWS のデータセット中に含まれる例

snippet1	In 1955 the Soviet Union forwarded \$ 100 million in <i>credit</i> to Afghanistan, which financed public transportation, airports, etc.
snippet2	Only congress has the authority to coin this <i>money</i> that should be used by the States.

## TOPIC [12]

各単語にトピックごとの分散表現を割り当てる。

CBOW および SGNS はベースラインである。提案手法のバリエーションとして, Paetzold and Specia [11] から着想を得て, 各単語と `context-word` の品詞を用いる手法も実装, 比較評価を行う。例えば, 単語 `soft` の `context-word` が `cheese` の場合, 品詞を用いる評価では `soft_adj_cheese_n` の分散表現を評価する。

## 5. 単語間の意味的類似度推定タスク

多義性を考慮する分散表現のテキスト解析における性能を評価するために, Stanford Contextual Word Similarity (SCWS) のデータセット<sup>\*2</sup> [14] を利用する。このデータセットでは, 2 つのターゲット単語とそれらの出現する文脈 (前後 50 単語) が与えられる。タスクは, 所与の文脈中でのターゲット単語間の意味的類似度を推定することである。SCWS データセットは, Amazon Mechanical Turk を用いてアノテータを採用し, 2,003 組の単語対のそれぞれに 10 人が類似度を付与している。表 1 に示すように, 特定の文脈において意味的に近い単語対や, 逆に特定の文脈において意味的に遠い単語対が含まれるため, このタスクでは単語の多義性を考慮して類似度を推定する必要がある。

### 5.1 意味的類似度の推定

まず, ターゲット単語  $w_i$  および  $w_j$  における `context-word`  $c_i \in C_i$  および  $c_j \in C_j$  を依存構造解析によって抽出する。次に, これらを用いて各ターゲット単語の分散表現のセット  $V_i$  および  $V_j$  を得る。そして, 以下の 3 つの方法で, 所与の文脈中でのターゲット単語間の意味的類似度を推定する。

$$S_{\text{avg}} = \frac{1}{|V_i||V_j|} \sum_{v_i \in V_i, v_j \in V_j} \cos(v_i, v_j) \quad (1)$$

$$S_{\text{max}} = \max_{v_i \in V_i, v_j \in V_j} \cos(v_i, v_j) \quad (2)$$

$$S_{\text{min}} = \min_{v_i \in V_i, v_j \in V_j} \cos(v_i, v_j) \quad (3)$$

ただし,  $|\cdot|$  は集合の要素数を,  $\cos(\cdot, \cdot)$  は 2 つのベクトル間の余弦類似度を表す。

$S_{\text{avg}}$  では, それぞれの文脈に出現する全ての `context-word` を等しく考慮して, 所与の文脈中での単語間の意味的類似度を推定する。 $S_{\text{max}}$  および  $S_{\text{min}}$  では, それぞれの

<sup>\*1</sup> <https://dumps.wikimedia.org/enwiki/20170601/>

<sup>\*2</sup> <http://www-nlp.stanford.edu/~ehhuang/SCWS.zip>

表 2 SCWS データセットにおけるスピアマンの順位相関係

Model	Spearman's $\rho$
SGNS [12]	0.59
MSSG [12]	0.61
TOPIC [12]	0.61
<hr/>	
CBOW	0.63
POS	0.64
Ours ( $S_{avg}$ )	0.64
Ours ( $S_{max}$ )	0.64
Ours ( $S_{min}$ )	0.63
Ours+POS ( $S_{avg}$ )	0.65
Ours+POS ( $S_{max}$ )	0.64
Ours+POS ( $S_{min}$ )	0.65

文脈に出現する context-word の重要度は異なると仮定し、代表的な 1 組のみを考慮して、所与の文脈中での単語間の意味的類似度を推定する。

ここで、context-word が存在しない場合および context-word によって与えられる分散表現が語彙に存在しない場合、類似度を計算できない。これらの場合は、事前学習した CBOW の単語分散表現を用いて、単純にターゲット単語間の余弦類似度を計算するものとする。

## 5.2 実験結果

表 2 に、モデルが推定した類似度と人手で定義された類似度のスピアマンの順位相関係数を示す。SGNS, MSSG, TOPIC の結果は文献 [12] で報告されている値である。CBOW [2] および POS [11] は、提案手法と同じ学習データおよび実験設定で我々が再実装したものである。Ours は単語のみ、Ours + POS は単語とその品詞を用いる提案手法である。表 2 の結果から、context-word を用いるモデルが context-word を用いないモデルよりもわずかに高い精度を示すことが分かる。

表 3 は、context-word によって与えられる分散表現がどれだけ語彙に含まれているかを調べたものである。品詞情報を付与した場合でも、8 割以上の分散表現が語彙に含まれていることが分かる。

## 6. 語彙的換言タスク

多義性を考慮する分散表現のテキスト生成における性能を評価するために、LS-SE と LE-CIC の 2 つのデータセット\*3を利用する。これらのデータセットでは、ターゲット単語と文脈に加えて、言い換え候補群が与えられる。タスクは、所与の文脈を考慮しつつ、ターゲット単語との置換可能性の観点から言い換え候補群をランク付けすることである。そのため、このタスクも文脈中での単語の意味を考慮する必要がある。

\*3 <https://github.com/stephenroller/naacl2016>

表 3 意味的類似度推定タスクにおける語彙の網羅性

	In vocabulary / All tokens	
Ours	7,297 / 8,772	83.2%
Ours+POS	7,058 / 8,772	80.5%

### LS-SE [15]

SemEval-2007 の Lexical Substitution タスクで用いられたデータセット。201 種類のターゲット単語について、それぞれ 10 種類の文脈が与えられている。各ターゲット単語には、5 人のアノテータが最大 3 種類ずつの言い換えを文脈を考慮して付与している。

### LS-CIC [16]

Lexical Substitution のための大規模なデータセット。15,629 語のターゲット単語について、6 人のアノテータが最大 5 種類ずつの言い換えを文脈を考慮して付与している。LS-SE とは異なり、文脈としてターゲット単語が出現する文に加えてその前後の文も与えられる。

先行研究 [9, 12, 19] に従い、各ターゲット単語に付与された全ての言い換えの和集合を言い換え候補群として用いる。そのため、これらの候補は少なくともある文脈においてはターゲット単語と意味的および構文的に十分に近い単語であり、モデルは文脈を用いてターゲット単語の語義を区別する必要がある。

人手で定義された言い換えランキングと、システムが推定した言い換えランキングは、Generalized Average Precision (GAP) [20] を用いて評価される。GAP は語彙的換言タスクで広く用いられる評価尺度であり、正解事例の重みを考慮してランキングを評価できる。本タスクでは、各言い換え候補を付与したアノテータの人数が重みとなる。

### 6.1 言い換え候補のランキング

このタスクでは文脈  $s$  を考慮してターゲット単語  $w_t$  と言い換え候補  $w_p$  の置換可能性を推定する。本実験では、ターゲット単語と言い換え候補の分散表現間の類似度が高いほど、その言い換え候補が高い置換可能性を持つと仮定する。まず、ターゲット単語  $w_t$  の context-word  $c_t \in C_t$  を依存構造解析によって抽出する。次に、これらを用いてターゲット単語  $w_t$  の分散表現のセット  $V_t$  を得る。続いて、ターゲット単語の各 context-word を用いて、言い換え候補  $w_p$  の分散表現のセット  $V_p$  を得る。そして、以下の 2 つの方法で、ターゲット単語の分散表現  $v_p \in V_p$  と言い換え候補の分散表現  $v_c \in V_c$  の間の類似度を計算し、言い換え候補群をランキングする。

#### Cos

ベクトル間の余弦類似度  $\cos(v_t, v_p)$  を計算する。

#### balAddCos [9]

この類似尺度では、ターゲット単語と言い換え候補の類似度だけでなく、言い換え候補と文脈の類似度も考

表 4 LS-SE データセットおよび LS-CIC データセットにおける GAP スコア

Model	LS-SE		LS-CIC	
	Cos	balAddCos	Cos	balAddCos
SGNS [12]	40.5	40.9	32.1	36.1
MSSG [12]	41.1	NA	37.8	NA
TOPIC [12]	NA	42.8	NA	40.9
CBOW	41.0	40.1	44.1	44.4
POS	41.8	42.1	46.5	46.7
Ours ( $S_{avg}$ )	47.0	47.2	45.7	45.9
Ours ( $S_{max}$ )	<b>48.4</b>	<b>48.5</b>	46.4	46.5
Ours ( $S_{min}$ )	45.0	31.1	44.4	44.6
Ours+POS ( $S_{avg}$ )	46.7	47.0	45.7	45.9
Ours+POS ( $S_{max}$ )	47.7	48.1	46.3	46.5
Ours+POS ( $S_{min}$ )	45.4	30.6	45.3	45.5
Ours ( $S_{avgc}$ )	47.4	47.7	46.1	46.2
Ours ( $S_{maxc}$ )	<b>48.4</b>	<b>48.5</b>	46.4	46.5
Ours ( $S_{minc}$ )	45.5	30.4	45.2	45.4
Ours+POS ( $S_{avgc}$ )	46.9	47.4	47.7	47.8
Ours+POS ( $S_{maxc}$ )	47.7	48.0	<b>48.1</b>	<b>48.2</b>
Ours+POS ( $S_{minc}$ )	45.5	30.1	46.9	47.1

表 5 語彙的換言タスクにおける出力例：ターゲット単語は **bold** で、context-word は *italic* で表示. 出力は言い換え候補群のランキングで、括弧内の数値は各候補の重みを表す.

入力	... you are carrying on two conversations at once and you are required to <i>listen</i> <b>hard</b> .
出力	carefully (4), intensively (0), closely(1), intently(1), seriously (0), ...
入力	One event in particular <i>hits</i> the platoon <b>hard</b> : the death of its platoon leader, ...
出力	badly (3), heavily (0), strongly (0), severely (1), firmly (0), ...

慮し、 $|s| \cos(v_t, v_p) + \sum_{w_i \in s} \cos(v_p, w_i)$  を計算する.

複数の context-word が存在する場合、最終的な置換可能性を推定する単純な方法は、5.1 節で説明した  $S_{avg}$ ,  $S_{max}$  および  $S_{min}$  を用いることである. しかし、意味的類似度推定タスクとは異なり、本タスクではターゲット単語と言い換え候補群との文脈が共通のため、共通の context-word のみを考慮して、より正確に文脈を考慮することができる. そこで本実験では、 $S_{avg}$ ,  $S_{max}$  および  $S_{min}$  に加えて、共通の context-word のみを考慮する以下の  $S_{avgc}$ ,  $S_{maxc}$  および  $S_{minc}$  によって、所与の文脈を考慮しつつ言い換え候補をランキングする.

$$S_{avgc} = \frac{1}{|C_t|} \sum_{c_t \in C_t} \text{sim}(\text{vec}(w_t, c_t), \text{vec}(w_p, c_t)) \quad (4)$$

$$S_{maxc} = \max_{c_t \in C_t} \text{sim}(\text{vec}(w_t, c_t), \text{vec}(w_p, c_t)) \quad (5)$$

$$S_{minc} = \min_{c_t \in C_t} \text{sim}(\text{vec}(w_t, c_t), \text{vec}(w_p, c_t)) \quad (6)$$

ここで、 $\text{sim}(\cdot, \cdot)$  には  $Cos$  または  $balAddCos$  を用いる. また、 $\text{vec}(\cdot, \cdot)$  では、context-word を用いて単語の分散表現を得る. これらの手法で推定された置換可能性のスコアを用いて、スコアの高い順に言い換え候補群をランキングする.

## 6.2 実験結果

表 4 に、LS-SE データセットおよび LS-CIC データセットにおける GAP スコアを示す. 類似尺度として  $Cos$  を使用したとき、全ての提案手法が先行研究よりも高い性能を発揮することが分かる. LS-SE では既存手法である *MSSG*, *POS*, *TOPIC* に比べ、それぞれ 7.3 ポイント、6.4 ポイント、5.7 ポイント高い GAP スコアを達成した. また LS-CIC ではそれぞれ 10.3 ポイント、1.5 ポイント、7.3 ポイント高い GAP スコアとなった. これらの結果から、context-word を用いて単語に複数の分散表現を割り当てる提案手法が、トピックや品詞を用いる既存手法に比べて効果的に語義を捉えることができると言える. 表 5 には、context-word を用いて多義語 **hard** の語義を捉えることに成功した例を示す. 最も多くのアノテータによって付与された言い換え候補を最上位にランク付けできている.

類似尺度として  $balAddCos$  を使用したときも、多くの提案手法が先行研究の性能を上回ることが分かる. しかし、その効果は分散表現の生成手法およびデータセットに依存することが分かる.  $balAddCos$  では文脈に現れる全ての単語との類似度を考慮するため、その有効性は文によって変化すると考えられる. 提案手法においては、 $Cos$  を用いるこ

表 6 語彙的換言タスクにおける語彙の網羅性

		Coverage
LS-SE	Ours	82.2%
	Ours+POS	79.7%
LS-CIC	Ours	73.4%
	Ours+POS	71.7%

とで安定して高い性能を達成した。これは提案手法では依存構造を用いて抽出した context-word により、意味表現において重要な文脈の情報を捉えられているためと考えられる。

また、複数の context-word を利用できる時、 $S_{maxc}$  が最も高い性能を示している。これらの結果から、同一の context-word によって与えられる分散表現を用いることで、全ての context-word を考慮するよりも高精度に単語間の置換可能性を推定できると言える。 $S_{avgc}$  の性能が  $S_{maxc}$  に及ばないのは、各分散表現の影響が平均することによって軽視されるからである。ターゲット単語の語義をピンポイントで表現できる context-word が存在したとしても、平均することによってその影響が薄れてしまう。また、 $Minc$  では類似度の下限値を基準にランク付けしている。 $Maxc$  であればその文中で使われている意味を表していない context-word との類似度は例え低くても無視されるが、 $Minc$  ではその影響を受けるため、精度が低くなったと考えられる。

LS-SE データセットと LS-CIC データセットの結果を比較すると、context-word を用いるときの GAP スコアの改善は、LS-SE データセットの方が大きいことが分かる。これは、context-word により与えられる分散表現が語彙に含まれるか否かに依存する。表 6 に示すように、LS-SE データセットにおける語彙の網羅性は LS-CIC データセットにおける語彙の網羅性よりも約 9% 高い。語彙の網羅性が高い LS-SE データセットでは、context-word を用いる影響がより大きく、本タスクの性能がより大きく改善されると考えられる。

次に、学習データにおける context-word の出現回数が、分散表現の学習にどのような影響を与えているのか検証した。ここでは、両方のデータセットで高い GAP スコアを示した  $Ours(S_{maxc})$  における  $balAddCos$  の類似尺度を用いる。検証のため、新たに 2 つのモデルを構築した。1 つは学習データにおいて出現頻度が 5 回以下の context-word のみを用いた分散表現を使用する。もう 1 つは出現頻度が 100 回以上の context-word のみを用いる。表 7 に、LS-SE データセットおよび LS-CIC データセットにおける各モデルの GAP スコアを示す。ここで、各単語に唯一の分散表現を生成する CBOW、単語と品詞を対にして分散表現を生成する POS と比較した結果から、出現頻度が 5 回以下の低頻度な context-word のみを使用した場合でも、GAP

表 7 context-word の出現頻度が与える影響

		LS-SE	LS-CIC
CBOW		40.1	44.4
	Less than 5	41.6	44.3
Ours	More than 100	43.3	45.6
	All	<b>48.5</b>	<b>46.5</b>
POS		42.1	46.7
	Less than 5	42.8	46.7
Ours+POS	More than 100	43.6	47.6
	All	<b>48.0</b>	<b>48.2</b>

スコアが向上していることが分かる。つまり、提案手法では事前学習によってデータスパースネス問題を軽減できていると言える。また、出現頻度が 100 回以上の高頻度な context-word を用いて訓練された分散表現は、より大きく GAP スコアを改善した。さらに、全ての context-word を考慮すると、GAP スコアは大幅に改善されることが示された。

## 7. まとめ

本研究では、語義を捉えるために 1 つの単語に複数の分散表現を割り当てる手法を提案した。提案手法では、依存関係にある context-word によって語義を捉えられると考え、context-word ごとに分散表現を生成した。単語間の意味的類似度推定タスクおよび語彙的換言タスクの評価により、我々の提案手法の有効性を確認できた。提案手法は特に語彙的換言タスクにおいて既存手法の性能を顕著に上回る性能を示した。また、事前学習によってデータスパースネス問題に対処でき、低頻度な context-word を効果的に利用しつつ語義を捉える分散表現を生成できることがわかった。

今後の課題は、1 つの context-word ではなく、文脈全体を考慮することである。これは、双方向のリカレントニューラルネットワークを使用することで実現できるだろう。また、本研究を複単語表現やフレーズにも拡張したい。

## 8. 謝辞

本研究は KDDI 財団による助成を受けたものである。

## 参考文献

- [1] Harris, Z. S.: Distributional Structure, *Word*, pp. 146–162 (1954).
- [2] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *Proceeding of the International Conference on Learning Representations*, pp. 1–12 (2013).
- [3] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *In Advances in Neural Information Processing Systems 26*, pp. 3111–3119 (2013).
- [4] Pennington, J., Socher, R. and Manning, C.: Glove:

- Global Vectors for Word Representation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543 (2014).
- [5] Levy, O. and Goldberg, Y.: Dependency-Based Word Embeddings, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 302–308 (2014).
- [6] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T.: Enriching Word Vectors with Subword Information, *Transactions of the Association for Computational Linguistics*, pp. 135–146 (2017).
- [7] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to Sequence Learning with Neural Networks, *In Advances in Neural Information Processing Systems*, pp. 1–9 (2014).
- [8] Mikolov, T. and Com, T. G.: Distributed Representations of Sentences and Documents, *Proceedings of the 31st International Conference on Machine Learning*, pp. 1188–1196 (2014).
- [9] Melamud, O., Levy, O. and Dagan, I.: A Simple Word Embedding Model for Lexical Substitution, *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pp. 1–7 (2015).
- [10] Neelakantan, A., Shankar, J., Passos, A. and McCallum, A.: Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space, *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1059–1069 (2014).
- [11] Paetzold, G. H. and Specia, L.: Unsupervised Lexical Simplification for Non-Native Speakers, *In Proceedings of AAAI Conference on Artificial Intelligence*, pp. 3761–3767 (2016).
- [12] Fadaee, M., Bisazza, A. and Monz, C.: Learning Topic-Sensitive Word Representations, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 441–447 (2017).
- [13] Athiwaratkun, B. and Wilson, A. G.: Multimodal Word Distributions, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1645–1656 (2017).
- [14] Huang, E. H., Socher, R., Manning, C. D. and Ng, A. Y.: Improving Word Representations via Global Context and Multiple Word Prototypes, *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 873–882 (2012).
- [15] McCarthy, D. and Navigli, R.: SemEval-2007 Task 10: English Lexical Substitution Task, *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 48–53 (2007).
- [16] Kremer, G., Erk, K., Padó, S. and Thater, S.: What Substitutes Tell Us - Analysis of an “All-Words” Lexical Substitution Corpus, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 540–549 (2014).
- [17] Li, J. and Jurafsky, D.: Do Multi-Sense Embeddings Improve Natural Language Understanding?, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1722–1732 (2015).
- [18] Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. and McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pp. 55–60 (2014).
- [19] Roller, S. and Erk, K.: PIC a Different Word : A Simple Model for Lexical Substitution in Context, *The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1121–1126 (2016).
- [20] Kishida, K.: Property of Average Precision and its Generalization: An Examination of Evaluation Indicator for Information Retrieval Experiments, *NII Technical Reports*, pp. 1–19 (2005).