単語の概念関係と分散表現の関係性

田中　雄也†　　田島　敬史††

† 京都大学大学院情報学研究科　〒 606-8501 京都府京都市左京区吉田本町
†† 京都大学大学院情報学研究科　〒 606-8501 京都府京都市左京区吉田本町
E-mail: †yuya.tanaka@dl.soc.i.kyoto-u.ac.jp, ††tajima@i.kyoto-u.ac.jp

あらまし　単語には，上位語や下位語，類義語といった概念関係を辞書化したものとして，シソーラスが知られている．シソーラスはほぼ手動で構築されているため，その構築コストは膨大であり，全ての単語やその概念関係を網羅することは難しい．特に，新しい単語やその用法の関係への対応には時間を要する．一方で単語の分散表現の獲得コストは，シソーラスの構築ほど大きくはなく，これを利用した演算もコストは小さい．さらに，分散表現の獲得に Web ページなどを用いることにより，新しい単語・用法への対応も容易である．よって，単語の分散表現から，それらの単語間の概念関係を推定できれば，シソーラスで網羅できないような語の概念関係もより容易に獲得できる．そこで，本論文では，単語の分散表現の類似度やノルム，空間的関係などを分析することで，単語間の概念関係と，単語の分散表現との間の関係を分析する．

キーワード　WordNet，概念関係，Word2Vec，分散表現

1　はじめに

自然言語処理や情報検索の分野において，シソーラスや分散表現の重要性は高まっている．シソーラスとは，単語の上下関係や部分・全体関係，対義関係，類義関係といった概念関係によって，単語を体系的にまとめた辞書のことである．シソーラスが効果的に用いられる場合は，例えば情報検索の分野においては，クエリ改善・緩和が考えられる．単純な Web 検索システムでは，“京都 お茶” というクエリが与えられると，“京都” と “お茶” の両語が含まれているような Web ページを返す．しかしこの場合，“お茶” が示す意味には様々な可能性が存在している．例えば，“抹茶” や “緑茶”，“喫茶店” や “休憩” といったものが挙げられる．これらに関連する Web ページを取得するためには，“お茶” の類義語であるこれらの単語で，クエリ中の “お茶” を置換する必要がある．このような場合にシソーラスを検索することで，置換候補となる単語を得ることができる．

こうしたシソーラスとして，WordNet[1] [1] [2] が知られている．WordNet は，1986 年にプリンストン大学が始めたプロジェクトである．当初は英語版のみであったが，現在は各国の機関によって，日本語版 [2] やドイツ語版 [3] といった他言語版も開発されている．

しかし，こうしたシソーラスが，全ての単語の関係を網羅することは極めて困難である．特に，既存の語の用法の変化や新語といったものへの対応には，時間を要する傾向にある．加えられるエンティティの数は非常に膨大であるにも関わらず，主に手動で構築されていることによって構築コストが非常に大きくなるからである．

分散表現とは，単語や文書の意味を表現する低次元のベクトルのことである．分散表現はベクトル形式であるためコンピューターによる演算処理も容易に行うことができる．例えば，

$$\mathrm{Queen} = \mathrm{King} - \mathrm{Man} + \mathrm{Woman}$$

といった，意味的な四則演算が可能である．また，分散表現のベクトル空間は単語の意味を表現しているため，単語の意味的なクラスタリングや類似度計算も可能である．しかし分散表現ではなく文字列の状態では，このような演算を行うことは不可能である．また，単語の意味に関する情報も格納することが不可能であるため，クラスタリングなども行うことは難しい．

また，分散表現はシソーラスに比べて獲得コストが小さい．なぜなら，分散表現は手動ではなく機械学習によって獲得されるものだからである．そのため，モデル学習にいくらか時間は必要であるが，手動で構築するシソーラスと比較すると，その獲得コストは小さくなる．

さらに，モデル学習のコーパスを適切に選択することで，既存の語の用法の変化や新語への対応も容易になる．こうしたコーパスの一つとして Web ページが挙げられる．Web ページは，頻繁に追加・書き換えが行われている．そのため，新語や新用法が生まれてから Web ページに出現するまでの時間差は小さくなると考えられる．Web ページの取得から分散表現のモデル学習までは全て自動化可能であるため，手動で構築されるシソーラスに比べ，新語・新用法への素早い対応が可能になり，単語・用法の網羅性が大きい．

これらの関係性を調べるため，本研究では実験を行った．本実験では，WordNet 上で何らかの概念関係にある単語ペアについて Word2Vec のモデルから分散表現をそれぞれ抽出し，それらの間の関係性を，類似度やノルムといった指標から分析する．また，WordNet 上で概念関係が登録されていない単語ペアについても同様の分析を行い，二つの結果に統計的有意差が

1：https://wordnet.princeton.edu/
2：http://compling.hss.ntu.edu.sg/wnja/
3：http://www.sfs.uni-tuebingen.de/GermaNet/

存在するのかどうか調べる実験を行う．

本論文の構成は以下の通りである．2 章では，本研究の関連研究を紹介する．3 章では，本研究で使用する Word2Vec について説明する．4 章では，本研究で使用する WordNet について説明する．5 章では，本研究で行う実験の詳細について説明する．また，その実験の結果について記述し，結果を考察する．6 章では，本論文の結論を述べるとともに，今後の課題について説明する．

2 関 連 研 究

この節では，単語の分散表現や，WordNet における概念関係と分散表現の関係性に関する先行研究について記述する．

2.1 単語の分散表現に関する先行研究

単語の分散表現を得るための手法は，盛んに研究されている．Mikolov ら [3] は，1 層のニューラル・ネットワークを用いて，分散表現を得る Word2Vec と呼ばれる手法を提案した．Word2Vec には，CBOW と Skip-gram の 2 種類のモデルが存在している．Word2Vec は本研究でも使用していることから，これらのモデルの詳細については，3 章において記述している．Mikolov らの実験では，類似単語の検索タスクにおいて，以前の手法に比べて大幅に少ない計算コストで，精度を改善させることに成功した．

一方で，学習用コーパスに含まれていない単語や，含まれていたものの出現回数が著しく小さい単語については，分散表現を得られない．このような現象は語彙の豊富な言語で特に起こりうる．Bojanowski ら [4] は，Word2Vec における Skip-gram に基づく新たなモデルを提案することで，この問題を克服した．Bojanowski らの手法では，n-gram を 1 つの単語として扱い，分散表現を学習する．

また，Pennington ら [5] は，Glove と呼ばれる手法を提案した．Glove は，global matrix factorization と local context window を組み合わせた，新たな global log-bilinear regression モデルである．スパース行列全体（単語共起行列全体）や大規模コーパスの個々の context window ではなく，単語共起行列内の非ゼロ要素のみを学習することで，学習の効率性や単語の類似度タスクにおける精度が向上した．Levy ら [6] は，分散表現の学習で使用する文脈に単語同士の依存関係を加えることで，特に syntactic なタスクにおける性能の良い分散表現を獲得した．Yin ら [7] は，meta-embedding と呼ばれる手法を提案した．meta-embedding では，異なる分散表現を組み合わせることで，より良い分散表現を獲得する．Salle ら [8] は，LexVec と呼ばれる手法を提案した．LexVec では，正の自己相互情報量行列の重み付き因数分解を使うことで分散表現を得る．

これらの研究は，単語の分散表現を学習するアルゴリズムそのものを改良することで，より良い分散表現を得るという研究である．一方，こうした手法によって得られた分散表現を，特定のアルゴリズムによって変換することで分散表現を改良するという研究も行われている．Vilnis ら [15] は，Gaussian Embedding と呼ばれる手法を提案した．分散表現を低次元ベクトル空間ではなく，ガウス分布空間へマッピングすることで，コサイン類似度や内積よりも特徴を捉えることが可能になっている．Rothe ら [9] は，単語の分散表現が，その語彙素の分散表現の総和であることと，synset の分散表現が，その各要素の語彙素の分散表現の総和であることの二つを仮定し，単語の分散表現を入力とする autoencoder を作成した．この autoencoder によって，与えられた分散表現のベクトル空間をより精度の高いベクトル空間へと変換した．Nguyen ら [11] は，HyperVec と呼ばれる手法を提案した．HyperVec では，上位語との類似度が，その他の関係の語との類似度よりも大きくなるように，かつ，上位語の分散表現のノルムが下位語の分散表現のノルムよりも大きくなるように，分散表現を学習する．

このように汎用的な分散表現を学習する手法だけでなく，上位語検出に特化した手法も数多く研究されている．Chang ら [10] は，分散表現中の各要素が全て非負であるようなベクトルを生成することで，単語間の包含関係の情報を分散表現のベクトル内に追加した．Glavaš ら [12] は，Dual Tensor Model と呼ばれる手法を提案した．この手法は，テンソルを通して上位語・下位語の関係や，部分・全体の関係を捉えるニューラル・ネットワークであり，モデルの学習には上位語などしか必要としない．そのため，学習用コーパスに出現することが少ない固有名詞や新語などへの対応も可能である．Nickel ら [13] は，n 次元ユークリッド空間ではなく，n 次元ポアンカレ球内にベクトルを生成する Poincaré embedding と呼ばれる手法を提案した．Poincaré embedding では，単語の類似性のみならず，階層性も表現可能な分散表現を獲得する．Vulić ら [14] は，LEAR と呼ばれる手法を提案した．LEAR では，与えられたベクトル空間を，上位語・下位語関係や IS-A 関係を強調するように変換する．この変換時にはベクトル空間の他に，WordNet の階層情報が与えられ，この情報に近づくように分散表現のノルムなどを調整する．

2.2 WordNet と分散表現の関係性に関する先行研究

単語の概念関係と分散表現の関係性に着目した研究は，単語分散表現の学習に関する研究以外にも存在している．Lee ら [16] は，WordNet 上での意味的距離を考慮した，新たな単語間の類似度評価尺度を提案した．Lee らは，WordNet 上での単語間の距離が小さいほど意味的類似度は大きいと仮定し，Word2Vec や Glove といった分散表現のコサイン類似度に，WordNet 上での単語間距離を組み合わせた．Handler ら [17] は，Word2Vec による単語の分散表現の類似度と WordNet の上での意味的関係の関係性について実験・分析した．Handler らは，Reuter のコーパスから特徴的な単語 41,600 語を選択し，それら各語について Google 社が学習させた Word2Vec のモデル [4] によって分散表現のコサイン類似度を計算し，その上位 200 語を抽出した．そして，抽出された各語が WordNet 上でどのような意味的関係の語として登録されているか調査し，WordNet 上で

4：https://code.google.com/archive/p/word2vec/

の出現回数やコサイン類似度の分布，正規化したベクトル空間内での距離，synset の Jaccard 係数を基に分析を行った．分析の結果，Word2Vec による単語分散表現の類似度が高い単語集合には，上位語や類義語が多く含まれる可能性が高いということが明らかになった．Handler らの研究は本研究と同じように，Word2Vec による単語分散表現と WordNet による単語の概念関係との関係性について調べた研究である．この研究では，分散表現の類似度が高い単語組について WordNet 上での概念関係について調べている．しかし本研究では，WordNet 上での概念関係ごとに分散表現の関係性を調べる．

3 Word2Vec

この節では，提案手法において使用する Word2Vec [3] のアルゴリズムについて解説する．

Word2Vec には，周辺単語群から該当単語を推定する Continuous Bag-of-Words Model (CBOW) と，該当単語から周辺単語群を推定する Continuous Skip-gram Model (Skip-gram) の 2 種類のモデルが存在する．

3.1 CBOW

本節では CBOW の解説を行う．

CBOW は 3 層のニューラルネットワークである．文書中において連続する語を同時に入力し，出力における学習対象の語の確率が最大となるような N 次元のベクトル表現を学習する．以下ではこのニューラルネットワークの学習過程を，入力から順番に見ていくこととする．

コーパス内の語彙を $w_1, w_2, \ldots, w_V$，学習対象単語を w_i とする．学習対象単語の前後合わせて C 語を周辺語群とする．入力単語 w'_i は V 次元の One-hot ベクトル $\boldsymbol{x}_{w'_i}$ で表現される．入力は

$$\boldsymbol{X} = (\boldsymbol{x}_{w'_1}\ \boldsymbol{x}_{w'_2}\ \ldots\ \boldsymbol{x}_{w'_C}) \tag{1}$$

である．入力層 $\rightarrow$ 隠れ層の重み行列 $\boldsymbol{W}$ は $N \times V$ であり，隠れ層の入力は

$$\boldsymbol{W}\boldsymbol{X} = (\boldsymbol{v}_{w'_1}\ \boldsymbol{v}_{w'_2}\ \ldots\ \boldsymbol{v}_{w'_C}) \tag{2}$$

となる．隠れ層の出力は

$$\boldsymbol{h} = \sum_{i=1}^{C} \boldsymbol{v}_{w'_i} \tag{3}$$

である．隠れ層 $\rightarrow$ 出力層の重み行列を $V \times N$ の行列 $\boldsymbol{W'}$ とすると出力層の入力は

$$\boldsymbol{y} = \boldsymbol{W'}\boldsymbol{h} = \begin{pmatrix} \boldsymbol{v}_{w_1}^T \cdot \boldsymbol{h} \\ \boldsymbol{v}_{w_2}^T \cdot \boldsymbol{h} \\ \vdots \\ \boldsymbol{v}_{w_V}^T \cdot \boldsymbol{h} \end{pmatrix} \tag{4}$$

となり，出力は

$$\boldsymbol{z} = \mathrm{Softmax}(\boldsymbol{y}) \tag{5}$$

$$\begin{aligned} \boldsymbol{z}_i &= \frac{\exp(\boldsymbol{y}_i)}{\sum_{k=1}^{V} \exp(\boldsymbol{y}_k)} \\ &= p(w_i | w'_1, \ldots, w'_C) \end{aligned} \tag{6}$$

となる．

最後に，$p(w_i | w'_1, \ldots, w'_C)$ が最大となるように，$\boldsymbol{W}$ と $\boldsymbol{W'}$ を更新する．まず，交差エントロピー誤差関数 E を考えると

$$\begin{aligned} E &= -\log p(w_i | w'_1, \ldots, w'_C) \\ &= -\boldsymbol{y}_i + \log \sum_{k=1}^{V} \exp \boldsymbol{y}_k \end{aligned} \tag{7}$$

である．$\boldsymbol{W'} = r_{ij}$ と書くと

$$r_{ij}^{(new)} = r_{ij}^{(old)} - \eta \frac{\partial E}{\partial r_{ij}} \tag{8}$$

として $\boldsymbol{W'}$ が更新される．$\eta > 0$ は学習率である．また，$\boldsymbol{u}$ を One-hot ベクトルとすると

$$\frac{\partial E}{\partial \boldsymbol{W}} = \boldsymbol{u}\boldsymbol{b} \tag{9}$$

となるような $\boldsymbol{b}$ を用いて

$$\boldsymbol{v}_{w_i}^{(new)} = \boldsymbol{v}_{w_i}^{(old)} - \frac{1}{C}\eta \boldsymbol{b} \tag{10}$$

として $\boldsymbol{W}$ が更新される．

以上のサイクルを繰り返した後に得られる

$$\boldsymbol{v}_{w_i} = \boldsymbol{W}\boldsymbol{x}_{w_i} \tag{11}$$

が，w_i の特徴ベクトルとなる．つまり，CBOW において単語の意味を獲得することは，文脈中の単語から対象単語が現れる条件付き確率を最大化することと等しい．

3.2 Skip-gram

本節では Skip-gram の解説を行う．Skip-gram は CBOW を逆転させたモデルである．以下ではこのニューラルネットワークの学習過程を，入力から順番に見ていくこととする．

学習するベクトル表現は N とする．コーパス内の語彙を $w_1, w_2, \ldots, w_V$，学習対象単語を w_i とする．学習対象単語の前後合わせて C 語を周辺語群とし，入力単語 w_i は V 次元の One-hot ベクトル $\boldsymbol{x}_{w_i}$ で表現される．入力は $\boldsymbol{x}_{w_i}$ である．入力層 $\rightarrow$ 隠れ層の重み行列 $\boldsymbol{W}$ は $N \times V$ であり，隠れ層の出力は

$$\boldsymbol{h} = \boldsymbol{W}\boldsymbol{x}_{w_i} = \boldsymbol{v}_{w_i} \tag{12}$$

である．隠れ層 $\rightarrow$ 出力層の重み行列を $V \times N$ の行列 $\boldsymbol{W'}$ とすると出力層の入力は

$$\boldsymbol{y} = \boldsymbol{W'}\boldsymbol{h} = \begin{pmatrix} \boldsymbol{v}_{w_1}^T \cdot \boldsymbol{h} \\ \boldsymbol{v}_{w_2}^T \cdot \boldsymbol{h} \\ \vdots \\ \boldsymbol{v}_{w_V}^T \cdot \boldsymbol{h} \end{pmatrix} \tag{13}$$

となり，出力は

$$\boldsymbol{z} = \mathrm{Softmax}(\boldsymbol{y}) \tag{14}$$

$$z_c = \frac{\exp(y_c)}{\sum_{k=1}^{V}\exp(y_k)} = p(w'_c|wi) \quad (15)$$

となる．最後に，$p(w'_c|w_i)$ が周辺語群 $w'_1,\ldots,w'_c,\ldots,w'_C$ において最大となるように，$\boldsymbol{W}$ と $\boldsymbol{W'}$ を更新する．

CBOW と同様に交差エントロピー誤差関数 E を考えると

$$E = -\log p(w_i|w'_1,\ldots,w'_C) = -y_i + \log\sum_{k=1}^{V}\exp y_k \quad (16)$$

である．$\boldsymbol{W'} = r_{ij}$ と書くと

$$r_{ij}^{(new)} = r_{ij}^{(old)} - \eta\frac{\partial E}{\partial r_{ij}} \quad (17)$$

として $\boldsymbol{W'}$ が更新される．$\eta > 0$ は学習率である．また，第 i 次元の要素 $\boldsymbol{b}_i$ が

$$\boldsymbol{b}_i = \sum_{k=1}^{V}(\sum_{c=1}^{C}\frac{\partial E}{\partial \boldsymbol{y}_{ck}})\cdot r_{ij} \quad (18)$$

であるような $\boldsymbol{b}$ を用いて

$$\boldsymbol{v}_{w_i}^{(new)} = \boldsymbol{v}_{w_i}^{(old)} - \eta\boldsymbol{b} \quad (19)$$

として $\boldsymbol{W}$ が更新される．

以上のサイクルを繰り返した後に得られる

$$\boldsymbol{v}_{w_i} = \boldsymbol{W}\boldsymbol{x}_{w_i} \quad (20)$$

が，w_i の特徴ベクトルとなる．つまり，Skip-gram において単語の意味を獲得することは，語彙を，対象単語に対する正例と負例にクラスタリングする際の誤りを最小化することと等しい．

4 WordNet

本節では，本研究において使用する WordNet について説明する．WordNet は，人間の語彙記憶に関する心理言語学の理論に基づいて構築された大規模英語辞書であり，1986 年にプリンストン大学が始めたプロジェクトである．WordNet には，名詞，動詞，形容詞，副詞が収録されており，これらは全て，独立した概念を表現する synset に分類される．synset は概念関係や語彙関係によって相互に接続されているため，WordNet 上で近い距離に位置する単語は，意味的に非常に類似しているということを示している．

次に，WordNet 上で用いられている概念関係ついて説明する．

Synonymy
語同士が，同じような意味を持つ類義語関係のことである．例えば，“検索” と “調べる” が挙げられる．

Hypernymy
語 w が，語 v よりも一般的な広い概念を表現しているような上位語関係のことである．例えば，“漢字” に対する “文字” が挙げられる．

Hyponymy
語 w が，語 v よりも特定の狭い概念を表現しているような下位語関係のことである．例えば，“文字” に対する “漢字” が挙げられる．

Meronymy
語 w が，語 v の構成要素となっているような部分語関係のことである．例えば，“カメラ” に対する “シャッター” が挙げられる．

Holonymy
語 w が，語 v を含んでいるような全体語関係のことである．例えば，“シャッター” に対する “カメラ” が挙げられる．

Antonymy
語 w が，語 v と対を成すような対義語関係のことである．例えば，“座る” に対する “立つ” が挙げられる．

5 実　　験

本節では，本研究に際して行う実験について述べる．なお本実験の目的は，単語の概念関係と分散表現の関係性について調査することである．

5.1 データセット

本実験では，概念関係の辞書として WordNet[5] を使用する．WordNet の詳細は 4 章に記述しているためここでは省略する．また，分散表現を獲得するために Word2Vec を使用する．なお本実験では，Google 社が公開している学習済みモデル [6] を使用することとする．このモデルは，Google News[7] のコーパスを使用して学習されたものである．Google News のコーパスには，およそ 1,000 億語が含まれているが，そのうち約 300 万語句について，300 次元の分散表現が学習されている．

5.2 調 査 項 目

本実験では，WordNet 上での概念関係が，Word2Vec が生成する分散表現のベクトル空間上でどのように表されるのか調べる．本節では，分析対象とする概念関係と，ベクトル空間を分析するための指標について具体的に述べる．

5.2.1 対象とする概念関係

分析対象とする概念関係は次の通りである．各概念関係がどのようなものであるかは，4 章に記述しているためここでは省略する．

- 類義語関係
- 上位下位関係
- 部分全体関係

5.2.2 ベクトル空間上での指標

Word2Vec が生成する分散表現の関係を，以下の指標によって分析する．

- 類似度

5：https://wordnet.princeton.edu/download
6：https://code.google.com/archive/p/word2vec/
7：https://news.google.com

分散表現の類似度を以下の方法によって計算する．なお，以下の各指標の説明においては，N 次元ベクトル $\boldsymbol{x}$，$\boldsymbol{y}$ を考える．また，これらの第 i 次元の要素をそれぞれ x_i，y_i と書く．

– コサイン類似度

– KL ダイバージェンス

KL ダイバージェンスは，本来，二つの確率分布間の差異の尺度である．しかし，分散表現は確率分布ではないため，ここでは次の式によって，N 次元ベクトル $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ によって表現される分散表現を確率分布 $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ へ変換する．

$$\begin{aligned} \boldsymbol{z}_i &= \frac{1}{1 + \exp -\boldsymbol{x}_i} \\ \boldsymbol{y}_i &= \frac{\boldsymbol{z}_i}{\sum_k^N \boldsymbol{z}_k} \end{aligned} \tag{21}$$

- ノルム

以下に挙げるノルムの差を比較する．

– L1 ノルム

– L2 ノルム

– L∞ ノルム

- 距離

分散表現がベクトル空間上に示す点間の距離を計算する．使用する距離関数は以下の通りである．

– マンハッタン距離

マンハッタン距離は，N 次元ベクトル $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ と $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ について

$$\text{distance}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{N} |\boldsymbol{x}_k - \boldsymbol{y}_k| \tag{22}$$

と計算される．

– ユークリッド距離

ユークリッド距離は，N 次元ベクトル $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ と $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ について

$$\text{distance}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_{k=1}^{N} (\boldsymbol{x}_k - \boldsymbol{y}_k)^2} \tag{23}$$

と計算される．

– チェビシェフ距離

チェビシェフ距離は，N 次元ベクトル $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ と $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N)$ について

$$\text{distance}(\boldsymbol{x}, \boldsymbol{y}) = \max_k (|\boldsymbol{x}_k - \boldsymbol{y}_k|) \tag{24}$$

と計算される．

- エントロピー

概念関係にある語の分散表現について，エントロピーを計算する．これまでの指標では，2 単語間の関係性に関する指標であったが，エントロピーは 1 単語の指標である．ここでも，KL ダイバージェンスの場合と同様に，式 21 によって，分散表現を確率分布へ変換する．

5.3 実験の手順

本節では，本研究において行う実験の手順について説明する．本実験では，WordNet 上での概念関係と，Word2Vec による単語分散表現との関係性について分析する．そのため，まず初めに調査対象とする語 $w_1, \ldots, w_N$ を選択する．これを語群 W とする．なお，W については，WordNet に含まれている全単語から 10,000 語をランダムサンプリングすることによって得る．

NLTK[8] を用いて $w_i \in W$ $(i = 1, \ldots, N)$ について WordNet を検索することで，5.2.1 節に示した関係にある語 $v_1^i, \ldots, v_M^i$ を，関係ごとにそれぞれ抽出する．これを語群 V とする．そして，語 w_i と語 $v_j^i \in V$ $(j = 1, \ldots, M)$ の組について 5.2.2 で示した関係性を全て計算する．この計算では，gensim[9] を用いて Word2Vec のモデルを読み込み分散表現を得る．分散表現を用いた計算を全ての v_j について行い，5.2.1 節に示した関係ごとに集計する．語群 V の抽出から集計までの一連の流れを，全ての w_i について行う．ここで得られる集計結果は，WordNet 上で何らかの概念関係にある単語ペアについての結果である．さらに，この一連の流れを全単語ペア w_i, w_j $(\forall i, j)$ についても行う．これによって得られる集計結果は，全ての単語ペアについての結果であり，WordNet 上での概念関係性は考慮されていない．最後に，得られた二つの集計結果を比較する．

5.4 結　　果

本節では，実験結果について述べる．

図 1 は，ある単語と概念関係にある単語の分散表現のエントロピーを概念関係ごとに集計した結果である．分散表現のエントロピーは，概念関係によらず違いがないことが分かる．この結果より，概念関係によって特定の軸に高い値・低い値が集中するといったことはないと言える．

図 2 は，単語ペアにおける各単語の分散表現のコサイン類似度を概念関係ごとに集計した結果である．また，図 3 はその KL ダイバージェンスの値を概念関係ごとに集計した結果である．概念関係にある単語の分散表現の類似度が高いことが分かる．特に，類義語関係は，二つの単語の意味が似ている関係であるため，コサイン類似度の値が 1 付近に集中していることや，KL ダイバージェンスの値が 0 付近に集中していることは直感と一致している．また，上位下位関係や部分全体関係についても，概念関係のない場合に比べると，分散表現が類似していることを示している．これは，上位下位関係や部分全体関係にある単語同士は，何らかの共通した意味を持っていることによると考えられる．しかし，概念関係のない場合に比べて分布の分散は大きい．これは，WordNet 上の単位距離が，実際の意味的距離と一致しておらず，単語によって意味的距離にばらつきが出てしまっていることが原因として考えられる．

図 4 は，単語ペアにおける各単語の分散表現の L1 ノルム差を概念関係ごとに集計した結果である．図 5，図 6 は，同様に，L2 ノルム差，L ∞ ノルム差を集計した結果である．また，図 7 は，単語ペアにおける各単語の分散表現のマンハッタン距離を概念関係ごとに集計した結果である．図 8，図 9 は，同様に，

8：https://www.nltk.org/

9：https://radimrehurek.com/gensim/

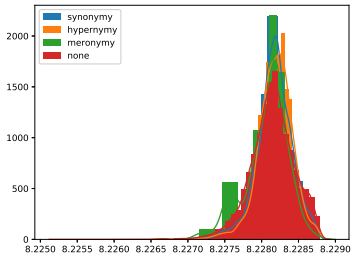

図 1　概念関係にある単語の分散表現のエントロピー

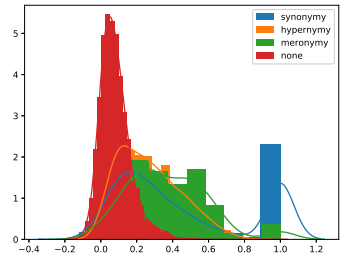

図 2　単語ペアのコサイン類似度

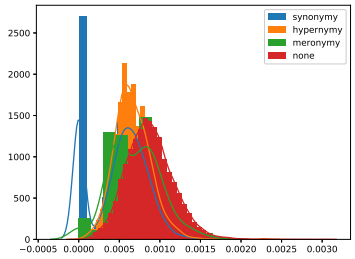

図 3　単語ペアの KL ダイバージェンス

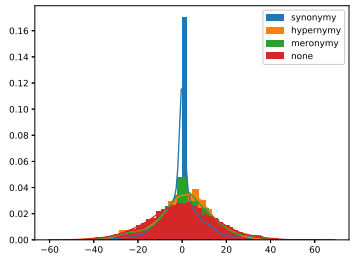

図 4　単語ペアの L1 ノルム差

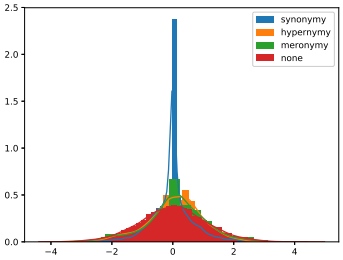

図 5　単語ペアの L2 ノルム差

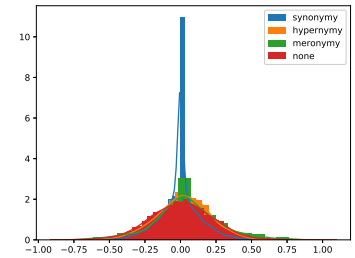

図 6　単語ペアの L∞ ノルム差

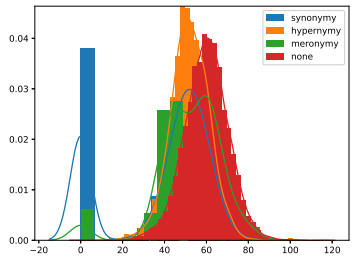

図 7　単語ペアのマンハッタン距離

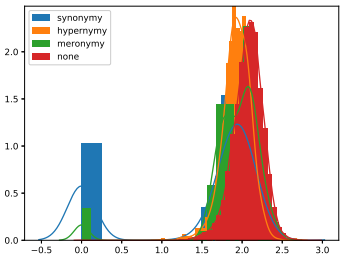

図 8　単語ペアのユークリッド距離

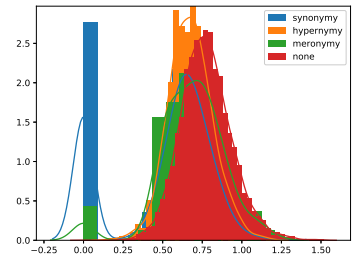

図 9　単語ペアのチェビシェフ距離

ユークリッド距離，チェビシェフ距離を集計した結果である．これらの図より，類義語関係の単語の分散表現間においては，距離やノルムが極めて近い値となっていることが分かる．類義語関係にある単語同士の意味は共通しているため，分散表現の空間において，非常に近い位置に各単語が割り当てられると考えられる．これに関しても，コサイン類似度や KL ダイバージェンスの場合と同様に直感に従っている．しかし，類義語関係以外の単語間において，分散表現間に差異はないことが分かる．

6　ま　と　め

本研究では，単語の概念関係と分散表現の関係性について調査した．そのために，WordNet 上で何らかの概念関係にある単語ペアについて Word2Vec のモデルから分散表現をそれぞれ抽出し，それらの間の関係性を，類似度やノルムといった指標から分析した．また，WordNet 上で概念関係が登録されていない単語ペアについても同様の分析を行い，両者を比較した．その結果，類義語関係にある単語の分散表現は，極めて近いものが生成されるのに対し，上位下位関係や部分全体関係にある単語ペアの分散表現間には，概念関係にない単語ペアと比較して有意な差異が見られなかった．

7　謝　　辞

本研究は，JST CREST (JPMJCR16E3)，JSPS 科研費 18H03245，JSPS 科研費 16K12430 の支援を受けたものである．

文　　献

[1] Miller, George A. ”WordNet: a lexical database for English.” Communications of the ACM 38.11 (1995): 39-41.
[2] Miller, George. WordNet: An electronic lexical database. MIT press, 1998.
[3] Mikolov, Tomas, et al. ”Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781 (2013).
[4] Bojanowski, Piotr, et al. ”Enriching word vectors with subword information.” arXiv preprint arXiv:1607.04606 (2016).
[5] Pennington, Jeffrey, Richard Socher, and Christopher Manning. ”Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.
[6] Levy, Omer, and Yoav Goldberg. ”Dependency-based word embeddings.” Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2014.
[7] Yin, Wenpeng, and Hinrich Schtze. ”Learning Word Meta-Embeddings.” Proceedings of the 54th Annual Meeting of

the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2016.

[8] Salle, Alexandre, Aline Villavicencio, and Marco Idiart. "Matrix Factorization using Window Sampling and Negative Sampling for Improved Word Representations." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2016.

[9] Rothe, Sascha, and Hinrich Schütze. "Autoextend: Extending word embeddings to embeddings for synsets and lexemes." arXiv preprint arXiv:1507.01127 (2015).

[10] Chang, Haw-Shiuan, et al. "Distributional Inclusion Vector Embedding for Unsupervised Hypernymy Detection." arXiv preprint arXiv:1710.00880 (2017).

[11] Nguyen, Kim Anh, et al. "Hierarchical embeddings for hypernymy detection and directionality." arXiv preprint arXiv:1707.07273 (2017).

[12] Glavaš, Goran, and Simone Paolo Ponzetto. "Dual tensor model for detecting asymmetric lexico-semantic relations." Association for Computational Linguistics, 2017.

[13] Nickel, Maximillian, and Douwe Kiela. "Poincaré embeddings for learning hierarchical representations." Advances in neural information processing systems. 2017.

[14] Vulić, Ivan, and Nikola Mrkšić. "Specialising word vectors for lexical entailment." arXiv preprint arXiv:1710.06371 (2017).

[15] Vilnis, Luke, and Andrew McCallum. "Word representations via gaussian embedding." arXiv preprint arXiv:1412.6623 (2014).

[16] Lee, Yang-Yin, et al. "Combining word embedding and lexical database for semantic relatedness measurement." Proceedings of the 25th International Conference Companion on World Wide Web. International World Wide Web Conferences Steering Committee, 2016.

[17] Handler, Abram. "An empirical study of semantic similarity in WordNet and Word2Vec." (2014).