Vol.2017-NL-233 No.17
2017/10/25

同義語を考慮した日本語の単語分散表現の学習

田口 雄哉[1,a)]　田森 秀明[1,b)]　人見 雄太[1,c)]　西鳥羽 二郎[2,d)]　菊田 洸[2,e)]

概要：近年，自然言語処理の研究において単語の分散表現が広く活用されている．word2vec などに代表される単語の分散表現は，分布仮説をもとに単語の分散表現を学習する．しかし，分布仮説にもとづいた学習を行なった場合，同義語や対義語に関わらず，同じ文脈に現れる単語は，似たようなベクトルになってしまうため，単語間の類似度を測る際に影響が出てしまう．その対策として，WordNet などの意味辞書から獲得した同義語対を用いて単語の分散表現を fine-tuning する手法が提案されているが，日本語での効果は報告されていない．そこで，本研究では，訓練済みの単語分散表現を用い，同義語対を用いた日本語の単語分散表現の fine-tuning を行なう．単語分散表現の評価は，日本語の単語類似度データセットを用いて行った．実験の結果，同義語対を考慮した学習手法を適用することで，既存の単語の分散表現よりも質が改善することを確認した．

1. はじめに

単語の分散表現は，多くの自然言語処理のタスクで活用されている．固定長の密ベクトルとして表現された単語分散表現は，その単語が出現する文脈を反映している．単語の分散表現の学習は，「ある単語の意味は，その単語と共起している単語によって特徴づけられる」という分布仮説 [10] にもとづいている．

しかし，分布仮説に基づいて分散表現の学習を行うと同義語のように似ているもののベクトル表現が似たものになる．一方で，「高い」「低い」のような対義語も共起する単語が似ているため意味が逆の単語でも，似たようなベクトルになってしまう．このような対義語対は，単語間の類似度が低い方が望ましいが，似たベクトルになってしまうため，自然言語処理のタスクにおいて分散表現を用いる際の課題となる．そのような事象に対処するために，Faruqui ら [7] は WordNet[15] などの意味辞書を用いて，単語の分散表現を，ある単語と意味的に関連している単語群は似たベクトルになるように fine-tuning する Retrofitting という手法を提案している．

本研究では，日本語における単語分散表現の Retrofitting の効果を検証する．具体的には，日本語 WordNet[11] を用いて同義語対を獲得し，Faruqui ら [7] の手法を用いて単語の分散表現の fine-tuning を行なう．単語の分散表現を評価には，Sakaizawa ら [19], [24] が公開している日本語の単語類似度データセットを用いた．実験の結果，日本語においても同義語対を用いて単語の分散表現の Retrofitting を行なうことで精度を確認した．

1 株式会社朝日新聞社
2 株式会社レトリバ
a) taguchi-y2@asahi.com
b) tamori-h@asahi.com
c) hitomi-y1@asahi.com
d) jiro.nishitoba@retrieva.jp
e) ko.kikuta@retrieva.jp

2. 関連研究

2.1 単語の分散表現

多くの単語分散表現は，分布仮説 [10] に基いている．そのため，単語ベクトルをどのように獲得するかについては様々な手法 [3], [6], [13], [14], [17] があるが，基本的には分布仮説に基いた学習を行っているため，異なる表層を持つ単語でも意味的に似ている単語は似たような値を持つベクトルが得られる．

word2vec などのような分布仮説にもとづいた単語分散表現の質を向上させるために，外部知識の活用がある．単語の分散表現の学習時に正則化として WordNet [15] などの外部知識を利用することで，分散表現の質を向上できるという報告がある [5], [12], [21].

単語の分散表現の学習時に外部知識を用いる方法の他に，Faruqui ら [7] は，WordNet [15] や FrameNet [2], PPDB [8] などの外部知識を用い，後処理として訓練済みの単語の分散表現を fine-tuninng する手法として Retrofitting を提案している．

Retrofitting は図 1 のように，意味的に関係のある単語間にエッジを引いたグラフとして表現し，関連する語同士のユークリッド距離を最小化することで単語の分散表現の

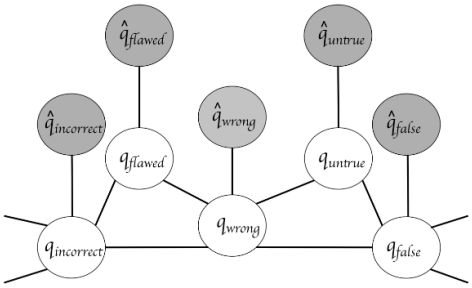

図 1 意味的に関連する語にエッジがある単語グラフ ([7] より抜粋)

最適化を行なう．WordNet などの一般的な単語の分散表現だけでなく，生体医学やサイバーセキュリティなど，特定のドメインの単語の分散表現にも適用することで，分散表現の質が向上すると報告されている [9], [18], [22].

2.2 日本語による単語分散表現の構築と評価

英語では，word2vec[*1] [13], [14] や Glove[*2] [17] といった手法を用いて訓練された単語の分散表現が公開され，よく用いられる一方で，日本語の単語の分散表現の場合は，日本語の Wikipedia から word2vec のツールを用いて単語の分散表現を構築することが多い．

日本語で公開されている単語ベクトルは，主に 3 種類ある．日本語の Wikipedia を用いた訓練済み分散表現としては，松田ら [26] が公開している word2vec による Wikipedia エンティティベクトル，Bojanowski ら [4] が公開している fastText ベクトルがある．また，Wikipedia 以外のコーパスで訓練されたものとして，浅原ら [25] は 258 億語からなる『国語研日本語ウェブコーパス』[1] を用い，word2vec のツールで実装されている Countinuous Bag-of-Words (CBOW) で学習された単語分散表現が nwjc2vec である．

吉井ら [27] は，word2vec [13], [14] と Glove [17] を用いた日本語の単語ベクトルの構築と，単語類推タスクと文完成タスクの 2 種類でその評価を行っている．単語類推タスクでは，表記ゆれや単語の活用形が与える影響によって，英語の単語類推タスクよりも正答率が低くなっていると報告している．

単語分散表現の性質を評価するタスクとして，単語類似度タスクがある．これは，単語ペアが与えられた際にその単語の類似度を計算し，類似度が高い順に単語ペアを並び替えるというタスクである．最終的に人手で評価された単語ペアの並びと，どれだけ相関があるかのスコアをスピアマンの順位相関係数によって算出する．日本語では，Sakaizawa ら [19], [24] は，日本語の単語ベクトルの評価のために，動詞，形容詞，名詞，副詞から成る単語類似度データセットを構築している．

[*1] https://code.google.com/archive/p/word2vec/
[*2] https://nlp.stanford.edu/projects/glove/

3. 単語分散表現の fine-tuning

本稿では，Faruqui ら [7] が提案している Retrofitting を用いて，単語の分散表現の fine-tuning を行なう．Retrofitting における目的関数は式 (1) の通りである．

$$\Psi(\mathbf{Q}) = \sum_{i=1}^{|\mathrm{V}|}\left[\alpha_i\|q_i - \hat{q}_i\|_2 + \sum_{(i,j)\in E}\beta_{i,j}\|q_i - q_j\|_2\right] \quad (1)$$

q_i は d 次元からなる単語ベクトルであり，全語彙の単語ベクトルを並べた行列は $\mathbf{Q} \in \mathbb{R}^{|\mathrm{V}|\times d}$ である．$|\mathrm{V}|$ は語彙数であり，d は単語の次元数である．この際，$q_i \in \mathbf{Q}$ が更新する単語ベクトルであり，$\hat{q}_i \in \hat{\mathbf{Q}}$ は初期値として用いる訓練済みの単語の分散表現とし，こちらは更新しない．$j:(i,j) \in E$ は i 番目の単語と関連する語[*3]である．α および β はハイパーパラメータである．

また，本実験では Faruqui ら [7] と同じく，式 (2) にもとづいて反復法で単語の分散表現を更新する．

$$q_i = \frac{\sum_{j:(i,j)\in E}\beta_{i,j}q_j + \alpha_i\hat{q}_i}{\sum_{j:(i,j)\in E}\beta_{i,j} + \alpha_i} \quad (2)$$

4. 実験

本実験では，日本語の単語の分散表現における Retrofitting の有効性を検証するために，以下の実験を行った．

4.1 評価方法

データとして，Sakaizawa らが公開している日本語単語類似度データセット [19], [24] で評価を行なう．単語の分散表現は，公開されている日本語の訓練済み単語分散表現に加え，word2vec [13], [14] と Glove[17] を用いて実験を行なう．評価は，人手でアノテーションされた単語の類似度と，単語の分散表現を用いたコサイン類似度を，スピアマンの順位相関係数によって行なう．

4.2 単語の分散表現

実験で用いる単語の分散表現には，鈴木ら [26] が公開している 200 次元の Wikipedia エンティティベクトル[*4] と Bojanowski ら [4] が公開している 300 次元の fastText ベクトル[*5]，および浅原ら [25] が公開している 200 次元の nwjc2vec[*6][25] を用いる．

公開されている訓練済みの分散表現に加え，word2vec[13], [14] および Glove[17] を用いて朝日新聞社の記事データか

[*3] 本稿では，WordNet から構築した同義語を用いる
[*4] http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/
[*5] https://github.com/facebookresearch/fastText/
[*6] 問い合わせにより入手可能．http://nwjc-data.ninjal.ac.jp/

表 1　朝日新聞記事データの統計

記事数	7,982,401(792 万記事)
文数	86,556,288(8655 万文)
単語数	2,361,591,403(23 億単語)

表 2　word2vec 実行時のパラメータ

CBOW or skip-gram	-cbow	{0, 1}
次元数	-size	300
文脈長	-window	8
負例サンプリング	-negative	5
階層化ソフトマックス	-hs	0
最低頻度閾値	-sample	1e-5
単語最低出現数	-min-count	3
反復回数	-iter	15

表 3　Glove 実行時のパラメータ

次元数	VECTOR_SIZE	300
文脈長	WINDOW_SIZE	8
単語最低出現数	VOCAB_MIN_COUN	3
反復回数	MAX_ITER	15

ら学習した単語の分散表現を用いる．朝日新聞の記事データの基礎統計は表 1 の通りである．分かち書きには MeCab-0.996[*7] と IPADIC-2.7.0 を用いた．

word2vec は公開されているスクリプト[*8]を用い，CBOW および Skip-gram モデルの訓練を行った．word2vec の訓練パラメーターは表 2 の通りである．

また，Glove も同様に公開されているスクリプト[*9]を用い，パラメーターは表 3 の通りである．

4.3　Retrofitting に用いる外部知識

Faruqui ら [7] が Retrofitting に用いる際に，WordNet [15] の同義語対を用いたのを倣い，本実験では，日本語版 WordNet [11] を用いる．

日本語版 WordNet の公式サイト[*10]にて提供されている人手で作成された同義語対[*11]に記載されている 11,753 同義語対を用いる．さらに，日本語版 WordNet のデータベースより自動で構築した 160,661 同義語対[*12]を用いた．

Retrofitting は式 (2) の反復法で行い，パラメーターは Faruqui ら [7] と同じく，反復回数を 10 回，$\alpha = 1, \beta = \mathrm{S}^{-1}$(S は更新する単語の同義語の数) とした．

4.4　評価詳細

Sakaizawa ら [19], [24] が公開している日本語類似度データセット[*13]を用いる．英語の単語ベクトルの評価とは異なり，データセット内には基本形でなく活用形でも記述されている．そのため，MeCab にて分かち書きをした際に，2 形態素以上ある場合は，単語間の類似度を求める際に単語の分散表現の合成を行なう必要がある．合成方法に関しては，堺澤ら [24] に従い，単語の分散表現 v は分かち書きされた N 個の単語の分散表現 $w_1, w_2, \cdots, w_N$ の平均を活用形で記された単語ベクトル v として定義する．これによって各単語ペア $(v1, v2)$ の分散表現を用いてコサイン類似度を計算し，類似度の高い順にソートする．それをもとに，人手でアノテーションされた類似度とのスピアマンの順位相関係数 (式 3) を計算する．D は対応する X と Y の順位であり，N は値のペア数である．

表 4　日本語類似度データセット

品詞	実際のデータ対	評価に用いたデータ対
動詞	1464	158
形容詞	960	93
名詞	1103	411
副詞	902	39
ALL	4429	701

$$\text{スピアマンの順位相関係数} = 1 - \frac{6\sum D^2}{N^3 - N} \qquad (3)$$

本実験では，朝日新聞の記事コーパスで学習された CBOW と skip-gram，Glove に加え，Wikipedia エンティティベクトル，fastText, nwjc2vec の 6 種類の単語分散表現の評価を行なう．各単語分散表現によって，語類似度データセットの単語が語彙のなかに含まれているものとそうでないものがある．そこで，本実験では，評価に用いるデータ対を揃えるため，表 4 にあるように，実験で用いる 5 種類の単語の分散表現全てに語彙が含まれているものだけを実験に用いた．

4.5　実験結果

実験結果を表 5 に記載する．実験を行った結果，全品詞 (ALL) と形容詞においては，スピアマンの順位相関係数が向上することが確認できた．これは，自動で構築したデータ対の量が多いため，人手によるデータ対を用いて Retrofitting を適用するよりも，評価データに現れる単語の分散表現の最適化に成功しているからだと考えられる．

一方で，Retrofitting を適用することによって質が悪化する場合もある．特に，副詞においては，自動で構築した同義語対を用いた場合，Skip-gram，Glove，そして fasttext においてスピアマンの順位相関係数が悪化している．理由としては，副詞の評価データの 24 語彙のうち，18 語彙が自動で構築した WordNet の同義語対に存在しているが，そ

[*7] https://taku910.github.io/mecab/
[*8] https://code.google.com/archive/p/word2vec/
[*9] https://github.com/stanfordnlp/GloVe
[*10] http://compling.hss.ntu.edu.sg/wnja/
[*11] 日本語 WordNet 同義語データベース ver.1.0
[*12] http://compling.hss.ntu.edu.sg/wnja/ にて提供されている「Japanese Wordnet and English WordNet in an sqlite3 database」を用い，訓練済みの単語ベクトル内にある語彙をもとにデータベースより同義語対を獲得した．
[*13] https://github.com/tmu-nlp/JapaneseWordSimilarityDataset

	動詞	形容詞	名詞	副詞	ALL
CBOW(朝日新聞コーパス)	37.7	38.0	27.2	15.0	24.8
CBOW(朝日新聞コーパス) + Retrofitting(人手)	36.7(+1.0)	38.0(±0)	32.4(+5.2)	18.9(+3.9)	29.8(+5.0)
CBOW(朝日新聞コーパス) + Retrofitting(自動)	**49.0**(+11.3)	54.6(+16.6)	34.6(+**7.4**)	34.1(+19.1)	33.5(+8.7)
Skip-gram(朝日新聞コーパス)	37.9	41.8	32.6	49.6	33.2
Skip-gram(朝日新聞コーパス)+ Retrofitting(人手)	35.6(+2.3)	42.0(+0.2)	37.1(+4.5)	50.7(+1.1)	39.0(+5.8)
Skip-gram(朝日新聞コーパス)+ Retrofitting(自動)	48.6(+10.7)	58.1(+16.1)	31.8(−0.8)	41.6(−8.0)	37.7(+4.5)
Glove(朝日新聞コーパス)	29.0	30.2	32.9	25.4	35.2
Glove(朝日新聞コーパス) + Retrofitting(人手)	29.0(±0)	30.2(±0)	37.2(+4.3)	27.2(+1.8)	39.6(+4.4)
Glove(朝日新聞コーパス) + Retrofitting(自動)	44.7(+15.7)	50.6(+**20.4**)	**39.0**(+6.1)	37.3(+**21.7**)	**44.2**(+**19.0**)
nwjc2vec[25]	36.0	55.4	32.4	43.4	29.4
nwjc2vec[25] + Retrofitting(人手)	34.2(−1.8)	55.5(+0.1)	37.9(+5.5)	47.9(+4.5)	33.7(+4.3)
nwjc2vec[25] + Retrofitting(自動)	48.3(+12.3)	**63.3**(+7.9)	35.9(+3.5)	42.4(−1.0)	36.1(+8.7)
Wikipedia エンティティベクトル [26]	35.4	28.5	29.4	47.4	25.3
Wikipedia エンティティベクトル [26] + Retrofitting(人手)	34.0(−1.4)	28.7(+0.2)	35.1(+5.7)	49.0(+1.6)	30.8(+5.5)
Wikipedia エンティティベクトル [26] + Retrofitting(自動)	41.4(+5.7)	52.3(+23.7)	33.0(+3.6)	**50.9**(+3.5)	32.3(+7.0)
fastText[4]	-7.4	3.7	22.1	24.6	23.2
fastText[4] + Retrofitting(人手)	-7.4(±0)	3.9(+0.2)	28.2(+6.1)	25.4(+0.8)	29.0(+5.8)
fastText[4] + Retrofitting(自動)	22.0(+**29.4**)	42.6(+**38.9**)	23.4(+1.3)	20.2(−4.4)	31.2(+7.9)

表 5 スピアマンの順位相関係数 × 100．スピアマンの順位相関係数が最も高かったものを太字で示しており，括弧内の太字はもとの訓練済み単語分散表現から最もスピアマンの順位相関係数が上昇したものを示している．

の中の 5 つの語彙全てがそれぞれの同義語対になっているため，どの単語もほぼ同じベクトルになってしまったことが原因と考えられる．

表 6 Retrofitting 前後での単語の最近傍

	嘆かわしい	少々	当然
更新前	情けない	みじん切り	筋違い
	腹立たしい	小さじ	やむを得ない
	あきれる	大さじ	べき
更新後	情けない	多少	自明
	むなしい	ちょっと	明白
	果敢ない	少し	無論

Retrofitting を適用することによって，ターゲットの単語の最近傍がどのように変化したかの一例を表 6 に示す．単語の分散表現には，朝日新聞コーパスで訓練した Skip-gram と，Retrofitting(自動) を適用した Skip-gram を用いた．

5. おわりに

本研究では，日本語 WordNet を用いて構築した同義語対を用い，Faruqui ら [7] が提案している Retrofitting を適用し，評価を行った．日本語の単語類似度データセットを用いて評価を行った結果，人手でアノテートされた同義語対と，WordNet から自動で構築した同義語対のどちらにおいても，訓練済みの単語の分散表現に Retrofitting を適用することで，日本語においても同義語対を考慮した単語の分散表現を獲得できていることが確認できた．

また，本実験では日本語版 WordNet を用いた同義語対の構築しか行わなかったが，他の外部知識を活用することも考えられる．梶原ら [23] は Bilingal Pivoting によって獲得した日本語の言い換え対を公開[*14]しており，Faruqui ら [7] の実験においても言い換え知識 PPDB を用いて Retrofitting を適用し，単語の分散表現の質が向上しているため，日本語でも同様に質の向上が見込める．また，Tamori ら [20] が朝日新聞記事の校正データを用いて，記事がどのように書き換えられたかの分析を行っている．校正データでは，ある単語がどのような単語に置換されたかをもとに，校正ログによる言い換えデータが獲得できる．これもどうように言い換え知識として，Retrofitting を適用することができる．

今後の課題は，Nikola ら [16] が提案している同義語と同時に対義語も考慮して単語の分散表現の Fine-tuning を行なうといった手法の適用が考えられる．しかし，日本語において，筆者らが知る限り，大規模な対義語対のデータは存在しない．そこで，今後は日本語の単語の分散表現を改善するために対義語データの構築を行なう．

また，本実験で用いた朝日新聞コーパスで訓練された単語ベクトルは公開する予定である．

参考文献

[1] Asahara, M., Maekawa, K., Imada, M., Kato, S. and Konishi, H.: Archiving and Analysing Techniques of the Ultra-large-scale Web-based Corpus Project of NINJAL, Japan, *Alexandria*, Vol. 25, No. 1-2, pp. 129–148 (2014).
[2] Baker, C. F., Fillmore, C. J. and Lowe, J. B.: The berke-

[*14] https://github.com/tmu-nlp/pmi-ppdb

ley framenet project, *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, pp. 86–90 (1998).

[3] Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C.: A neural probabilistic language model, *Journal of machine learning research*, Vol. 3, No. Feb, pp. 1137–1155 (2003).

[4] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T.: Enriching Word Vectors with Subword Information, *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146 (2017).

[5] Chang, K.-W., Yih, W.-t. and Meek, C.: Multi-Relational Latent Semantic Analysis, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1602–1612 (2013).

[6] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P.: Natural language processing (almost) from scratch, *Journal of Machine Learning Research*, Vol. 12, No. Aug, pp. 2493–2537 (2011).

[7] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E. and Smith, N. A.: Retrofitting Word Vectors to Semantic Lexicons, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, pp. 1606–1615 (2015).

[8] Ganitkevitch, J., Van Durme, B. and Callison-Burch, C.: PPDB: The Paraphrase Database., *HLT-NAACL*, pp. 758–764 (2013).

[9] Grouin, C., Hamon, T., Névéol, A. and Zweigenbaum, P.: Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis, *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis* (2016).

[10] Harris, Z. S.: Distributional structure, *Word*, Vol. 10, No. 2-3, pp. 146–162 (1954).

[11] Isahara, H., Bond, F., Uchimoto, K., Utiyama, M. and Kanzaki, K.: Development of the Japanese WordNet. (2008).

[12] Kiela, D., Hill, F. and Clark, S.: Specializing Word Embeddings for Similarity or Relatedness., *EMNLP*, pp. 2044–2048 (2015).

[13] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781* (2013).

[14] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems*, pp. 3111–3119 (2013).

[15] Miller, G. A.: WordNet: a lexical database for English, *Communications of the ACM*, Vol. 38, No. 11, pp. 39–41 (1995).

[16] Mrkšic, N., OSéaghdha, D., Thomson, B., Gašic, M., Rojas-Barahona, L., Su, P.-H., Vandyke, D., Wen, T.-H. and Young, S.: Counter-fitting Word Vectors to Linguistic Constraints, *Proceedings of NAACL-HLT*, pp. 142–148 (2016).

[17] Pennington, J., Socher, R. and Manning, C. D.: Glove: Global vectors for word representation., *EMNLP*, Vol. 14, pp. 1532–1543 (2014).

[18] Roy, A., Park, Y. and Pan, S.: Learning Domain-Specific Word Embeddings from Sparse Cybersecurity Texts, *arXiv preprint arXiv:1709.07470* (2017).

[19] Sakaizawa, Y. and Komachi, M.: Construction of a Japanese Word Similarity Dataset, *arXiv preprint arXiv:1703.05916* (2017).

[20] Tamori, H., Hitomi, Y., Okazaki, N. and Inui, K.: Analyzing the Revision Logs of a Japanese Newspaper for Article Quality Assessment, *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, Association for Computational Linguistics, pp. 46–50 (2017).

[21] Yu, M. and Dredze, M.: Improving Lexical Embeddings with Semantic Knowledge., *ACL (2)*, pp. 545–550 (2014).

[22] Yu, Z., Wallace, B. C., Johnson, T. and Cohen, T.: Retrofitting Concept Vector Representations of Medical Concepts to Improve Estimates of Semantic Similarity and Relatedness, *arXiv preprint arXiv:1709.07357* (2017).

[23] 梶原智之, 小町守, 持橋大地: Bilingual Pivoting による言い換え獲得の相互情報量に基づく一般化, **情報処理学会第** *231* **回自然言語処理研究会**, Vol. 2017, No. 21, pp. 1–8 (2017).

[24] 堺澤勇也, 小町守: 日本語動詞・形容詞類似度データセットの構築, **言語処理学会第** *22* **回年次大会** (2016).

[25] 浅原正幸, 岡照晃: nwjc2vec:『国語研日本語ウェブコーパス』に基づく単語の分散表現データ, **言語処理学会第** *23* **回年次大会** (2017).

[26] 鈴木正敏, 松田耕史, 関根聡, 岡崎直観, 乾健太郎: Wikipedia 記事に対する拡張固有表現ラベルの多重付与, **言語処理学会第** *22* **回年次大会** (2016).

[27] 吉井和輝, 中野幹生, 青野雅樹: 日本語単語ベクトルの構築とその評価, **情報処理学会第** *221* **回自然言語処理研究会**, Vol. 2015, No. 4, pp. 1–8 (2015).