複数粒度の分割結果に基づく日本語単語分散表現

真鍋 陽俊 [*1]　岡 照晃 [2]　海川 祥毅 [1]
髙岡 一馬 [1]　内田 佳孝 [1]　浅原 正幸 [2]
[1] 株式会社ワークスアプリケーションズ　[2] 人間文化研究機構 国立国語研究所
[*]manabe_h@worksap.co.jp

1 はじめに

単語の分散表現の学習は単語間の共起に基づいて行われる．しかし英語のように分かち書きする言語と違い，日本語は単語境界を明示しない分かち書きしない言語である．そのため，獲得される分散表現の語彙・ベクトルは，事前に行われる学習用コーパスへの分かち書きに大きく依存する．田口ら [1] は IPA 辞書，浅原ら [2] は UniDic に基づく単語分割で，それぞれ分散表現を構築・公開している．しかし，いずれも固有表現や複合語といった長い語を分かち書き時の辞書に登録していないため，固有表現や複合語の分散表現を獲得できていない．また固有表現や複合語を辞書登録していても，その長い語を構成するより短い語が学習時に別語扱いされると，それらの包含関係を反映した分散表現を獲得できない．例えば，「選挙管理委員会」のような長い語と，それを構成する「選挙」のような短い語の間には意味的な”関係”が存在しているはずであり, 獲得される分散表現にもその”関係”が反映されることが期待される．しかし彼らが採用している分散表現の学習手法 [3] では,1 語内のより細かな分割まで学習に使用できないため, 上記のような”関係”を分散表現には反映できない．

そこで本稿では, 複数粒度でのコーパス分かち書きを併用した日本語の単語分散表現の学習手法を提案する. コーパス分かち書きを行う際, 通常であれば 1 種類の分割粒度を採用し, 分散表現の学習に利用する. しかし本手法ではあらかじめ複数粒度での分かち書きを実施しておき, それらを同時に分散表現の学習用コーパスとする. これにより, 固有表現や複合語といった長い語をカバーしつつ, それらを構成する短い語との関係まで反映した単語分散表現が獲得できる．

提案手法で得た分散表現[*1] を使用し, 単語間類似度・文書分類タスクで比較評価と分析を行ったところ,

1. 固有表現のような長い語を幅広くカバーした分散表現獲得が有効であること
2. 複数の分割粒度を学習時に併用することで，長い語とそれを構成する短い語の間で高い類似性を獲得できること

を確認した．

2 分散表現の構築手法

2.1 既存の日本語単語分散表現の学習手法

分散表現の学習の多くは分布仮説に基づいている．この仮説では，ある 2 つの単語についてそれらが同様の文脈で用いられる場合，意味的に高い類似性を持つと仮定している．本稿では分散表現の学習アルゴリズムとして，Skip-gram with Negative Sampling(以下 SGNS)[3] を採用する．SGNS では，分かち書きされたテキストから収集された単語-文脈語ペア系列 $W = \left((w_t^{(1)}, w_c^{(1)}), ...(w_t^{(|W|)}, w_c^{(|W|)})\right)$ を学習用コーパスとする．ここで，w_t, w_c はそれぞれコーパスで観測された単語, 文脈語を表す．学習用コーパスは，通常，1 つの分割結果のみに基づく．そのため例えば，「/選挙管理委員会/」という分割が学習用コーパスに存在しても，その内部に包含されている「選挙」という語とはまったくの別語として学習が実行される．つまり，分割結果中に現れる長い語を構成するような短い語の情報までは，学習時に考慮しない．

2.2 提案手法: 複数粒度の分割を同時に用いた学習法

本稿では複数の分割粒度を同時に用いた分散表現の学習手法を提案する．固有表現のような長い語への分割，その中に含まれるより短い分割を学習用コーパスとして同時に用いることで，固有表現や複合語を語彙として幅広くカバーしつつ，より短い内部的な語の分割との関係も学習することを目指す．

複数分割の解析結果を得るために，我々は形態素解析器 Sudachi [4] を使用する．Sudachi は分割モードに応じて，短単位・中単位・長単位[*2]といった 3 種類の解析結果を得ることができる．例えば，”選挙管理委員会に立候補する”という入力文に対して，各分割モードで以下のような分かち書き結果を得る．

- **短単位**: 選挙/管理/委員/会/に/立/候補/する
- **中単位**: 選挙/管理/委員会/に/立候補/する
- **長単位**: 選挙管理委員会/に/立候補/する

この分割において各粒度ごとに個別に解析を行う必要はない．Sudachi の解析辞書内には，各エントリがどの分割単位であるか，長・中単位として登録される場合には内部的にどのような分割を持つか，といった情報が付与されている．そのため，長単位の解析結果からトップダウンに分割粒度を細かくしていくことができ，個別に解析を実施する際のように，各単位間で分割境界の不整合が起きることはない．例えば，**表 1** に示す通り，”選挙管理委員会”は長単位として登録されているが，付加情報として，中・短単位での内部的な分割情報も辞書内に保持されている．

提案手法ではこれら複数の分割候補を同時に扱う方法として，それぞれの単位で分かち書きした結果を 1 つの学習コー

[*1] 本稿の手法により構築された単語分散表現は後日公開予定である．

[*2] 以降，特に明記しない限り”長単位”は Sudachi の辞書内で長単位と定義される分割単位を指す．”中単位”, ”短単位”も同様である．

表 1　Sudachi 登録語が保持する構成語情報．Sudachi 登録語はより短い Sudachi 単位の情報を保持している．

見出し	Sudachi 単位	短単位分割情報	中単位分割情報
会	短単位	N/A	N/A
委員	短単位	N/A	N/A
委員会	中単位	委員/会	N/A
選挙管理委員会	長単位	選挙/管理/委員/会	選挙/管理/委員会

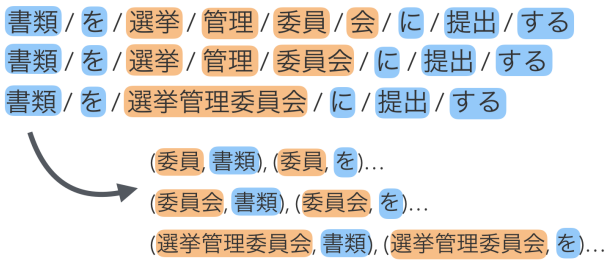

図 1　複数分割結果による単語-文脈語ペアの収集．複数分割結果を用いると，上記の例のように，”選挙管理委員会”と”選挙”など長い語とその内部の短い語 (橙色) の間で共通の文脈語 (水色) が観測されやすくなる．

パス $W_{multi} = W_A \cdot W_B \cdot W_C$ とする．(ここで W_A, W_B, W_C はそれぞれ，長・中・短単位の基準での分かち書き結果から収集した単語-文脈語ペア系列，$\cdot$ は系列の連結操作を表す．) こうすることでコーパスから収集される単語-文脈語ペアは図 1 のように，長単位とその内部の短・中単位の語で共通の文脈語を持つようになり，分散表現の学習の過程で両者の表現が近くなることが期待できる．結果，単語の長さ（単位）を超えて従来よりも単語間の類似性をとらえた分散表現の構築ができるようになる．

3　日本語単語分散表現の比較実験

3.1　分散表現の学習設定

本稿で作成する分散表現の学習用コーパスとして，nwjc[5] を使用した．nwjc はウェブ上のテキストをソースとして大規模な収集を行ったものであり，約 1 億のウェブページのテキストを含んでいる．テキストは nwc-toolkit[*3]によって日本語文抽出とテキスト正規化，および重複文の削除を前処理として実施した．nwjc は，UniDic 短単位で分割されたコーパスであるが，今回はその元データに Sudachi によって再度分かち書きを施した．Sudachi のバージョンは 0.1.1 を利用し， 解析辞書は 2018/07/04 時点のもので辞書 core と辞書 full の二種類をそれぞれ使用した．辞書 core は UniDic をベースに，実用性の観点から独自の基準で人名や地名のような固有表現語彙を追加し，前述の分割情報や生起コストなどの調整を重点的に行った辞書である．一方で辞書 core の登録基準を満たさなかった雑多な固有表現などは辞書 full にのみ登録されている．そのため，辞書 full が辞書 core を包含し語彙サイズが大きいものの，辞書 core の方がより精確に整備されている．

[*3] https://code.google.com/archive/p/nwc-toolkit/

本コーパスの統計量は**表 2** の通りであり，分散表現の学習時のハイパーパラメータを**表 3** に示す．

表 2　nwjc の基礎統計量 (辞書 core での解析時)

収集 URL 数	83,992,556	(8399 万)
文数	694,893,310	(6 億 9489 万)
長単位数 (のべ数)	10,178,505,823	(101 億 7850 万)

表 3　分散表現学習時のハイパーパラメータ

次元数	300
ウィンドウサイズ	8
負例サンプリング数	5
最低頻度閾値	1e-5
単語最低出現数	3
反復回数	15

本稿では nwjc を元に以下の分散表現を構築し比較を行った．

- **nwjc + core A**:
  短単位の分割結果のみ使用 (辞書 core)
- **nwjc + core C**:
  長単位の分割結果のみ使用 (辞書 core)
- **nwjc + core ABC**:
  短・中・長単位の分割結果を使用 (辞書 core)
- **nwjc + full ABC**:
  短・中・長単位の分割結果を使用 (辞書 full)

比較対象の単語分散表現として朝日新聞 SGNS[*4][1], nwjc2vec[*5][2] を使った．また本手法に類似して，長短語同士の包含関係でなく，1 語内の文字 n-gram を学習に利用する fastText[6] が公式に公開している日本語単語分散表現[*6]も比較に用いた．

なお以下の実験において，実際の評価データに適用する分かち書き手法は分散表現の学習時に使用されたものと同一のものを使用し，短・中・長単位を同時に表現学習したものについては長単位の解析モードを使用した．また未知語の扱いについて，fastText の場合は，内部の文字 n-gram の表現和に

[*4] http://www.asahi.com/shimbun/medialab/word_embedding/
[*5] http://nwjc-data.ninjal.ac.jp/
[*6] https://fasttext.cc/docs/en/crawl-vectors

表 4　単語間類似度タスクにおける実験結果 (スピアマンの順位相関係数)

	tmu				jwsan-1400	
	動詞	形容詞	名詞	副詞	類似度	関連度
朝日新聞 SGNS	**31.79**	35.51	25.39	**31.39**	47.46	62.43
nwjc2vec	17.82	36.66	24.29	25.88	52.85	61.55
fastText	18.19	33.60	29.77	29.84	**60.09**	66.03
nwjc + core A	24.65	39.40	26.88	23.29	48.05	62.00
nwjc + core C	25.09	39.88	29.85	24.09	51.85	64.14
nwjc + core ABC	26.28	**41.18**	30.49	25.62	52.55	65.20
nwjc + full ABC	25.79	40.55	**31.20**	25.60	53.85	**66.13**

よって表現獲得を行い，それ以外の手法については同次元数のゼロベクトルで代用した．

3.2　評価実験: 単語間類似度タスク

与えられた 2 つの単語ペアに対して人手アノテーションされた類似度と機械的に算出した類似度がどれほど相関があるのか各単語分散表現間で比較評価を行った．本稿では Sakaizawa ら [7](以下 tmu) と，猪原ら [8] (以下 jwsan-1400) によって公開されている評価データのの 2 つを使用する．総データ数はそれぞれ 4429 ペアと 1400 ペアであり，評価指標としてスピアマンの順位相関係数を使用した．

tmu の評価データには複数の語からなるエントリが多く含まれているため田口ら [1] にならい，各エントリに対して分かち書きを施し，各語の分散表現の平均和を取ることで表現獲得を行った．**表 4** に実験結果を示す．

全体的な傾向として，1 つの分割粒度のみを用いたものより複数の分割粒度を同時に学習に使用した方がより高い性能となることが確認できる．特に形容詞の精度向上は，様々な粒度で修飾先の名詞が文脈語として得られたことが要因ではないかと考えられる．また，jwsan-14000 内のエントリは Sudachi 短単位として登録されている語がほとんどであるが，nwjc + core A に比べて nwjc + core C の方が性能が高い傾向にあることから，長単位の分割結果を用いた方が，短単位のみで作成した分散表現より良い類似性を獲得できていることがわかる．これは短単位より長単位で分割することによって，意味的に関連する文脈語を拾いやすくなったためであると考えられる．

3.3　評価実験: 文書分類タスク

次に，実応用タスクとして，文書分類タスクによる評価実験を行った．評価データとして，livedoor コーパス[*7]を使用した．livedoor コーパスは全 7,367 文書，9 クラスからなる．

各文書の特徴量を作成するために，文書を形態素解析し，名詞と判定された語の分散表現について平均和を取ることで文書の特徴量ベクトルとした．この特徴量に基づき 1 対多のロジスティック回帰による分類器を構築し，10 分割交差検証で性能評価を行った．正則化項の強さは 1.0 とした．**表 5** に実験結果を示す．

複数分割を用いた学習の有無ではあまり大きな精度差は見られなかったが，辞書 core を使用するより辞書 full を用いた方が分類精度が高くなった．これは，より多様な固有表現を素性として活用できているためであると考えられる．また，比較対象の中で nwjc2vec が一番性能が高くなっているが，これは分散表現学習時のアルゴリズム・ハイパーパラメータの違いによるためだと考えられる (nwjc2vec では CBOW のアルゴリズム [2] による学習がなされており，また負例サンプリング数も 25 となっている)．

表 5　livedoor データセットにおける文書分類予測精度 (accuracy)

朝日新聞 SGNS	0.820 ±0.0013
nwjc2vec	**0.843** ±0.0010
fastText	0.810 ±0.0009
nwjc + core A	0.828 ±0.0014
nwjc + core C	0.831 ±0.0013
nwjc + core ABC	0.832 ±0.0013
nwjc + full ABC	0.838 ±0.0012

4　分析

4.1　固有表現のカバレッジ

今回獲得された分散表現がどれだけの固有表現をカバーしているのか確認を行った．評価対象の固有表現の表層系は拡張固有表現タグ付きコーパス [9] から取得[*8]した．各種分散表現についてどれほど固有表現がカバーされているのか**表 6** に示す．

短単位の分割結果に基づく場合であっても解析器の未知語処理などによりある程度の固有表現のエントリは存在しているが，特に辞書 full を解析器として使用しているものが従来の分散表現に比べ，幅広い固有表現をカバーしていることがわかる．

[*7] https://www.rondhuit.com/download.html#ldcc

[*8] 現代日本語書き言葉均衡コーパス（BCCWJ）にアノテートされている Name の大階層タグ全 11 種を持つエントリを対象に収集した．そのうち数値表現，空白記号，マスク記号を含むものと，URL, Email のタグを持つものは対象外とした．

表 6　各種分散表現の固有表現のカバー率 (n=23,354)

朝日新聞 SGNS	27.92%
nwjc2vec	28.55%
fastText	30.00%
nwjc + core A	29.35%
nwjc + core C	36.31%
nwjc + core ABC	36.99%
nwjc + full ABC	53.65%

4.2　語間の類似性

実際に学習された分散表現において，長単位とその内部に含まれる中単位の語の類似度に関する分析を行った．評価対象となる長単位と中単位の語のペアは Sudachi の辞書 core より収集した．各ペアの間の類似度分布を図 2 示す．

nwjc + core C と比較すると，nwjc + core ABC の方がより全体的な類似度が高くなった (それぞれ類似度の平均値は 0.4161 と 0.4620)．これは学習時に長単位とその内部の語で共通の文脈語が収集されたためだと考えられる．特に地名等については大きな類似度の向上があり，ある中単位の語の出現の多くが長単位の内部の語として現れるケースでこの現象が見られた．具体例を表 7 に示す．

表 7　複数分割粒度の学習による cosine 類似度の向上例

評価対象ペア	core C	core ABC
(埼玉県上尾市, 埼玉県)	0.562	0.686
(埼玉県上尾市, 上尾市)	0.628	0.795
(読解力, 読解)	0.798	0.822
(読解力, 力)	0.310	0.343

5　まとめと今後の展望

本稿では，固有表現や複合語を幅広くカバーしながら複数の分割候補を用いた分散表現の構築を実施した．実験結果より，複数分割粒度を同時に用いると各種タスクで性能向上が見られた．これに加えて，固有表現や複合語のような長い語が内部の語とより類似度が高くなるといった傾向が確認できた．

今後の方向性として，第一に，複数分割粒度の同時学習法をその他の分散表現学習手法や単語分割方法と組み合わせることが挙げられる．例えば，Bojanowsk ら [6] は語内部の文字 n-gram も同時に学習することで，より高品質な分散表現が構築できることを示している．本稿と同様に彼らも語の内部情報を用いており，詳細な比較の実施や本稿の手法との併用が考えられる．また，近年では byte pair encoding(BPE) など部分単語単位に基づいて分割器を構築する方法 [10] が存在し，今後は，このような手法により獲得される複数分割結果を使った分散表現の構築にも取り組んでいきたいと考えている．

第二に，未知語や新語に対する分散表現の取得が考えられる．特に固有表現などは日々新しい表現が増えており，長単位の語の内部の分割情報から意味構成されない事例も数多くある．これらの表現を効率的に獲得するための継続的なメンテナンス方法やオンライン学習法の策定が必要であると考えられる．また本稿の手法では通常に比べるとおよそ 3 倍の学習コーパス量となっているため，学習データの量と精度の関係等，結果のより詳細な分析も必要である．

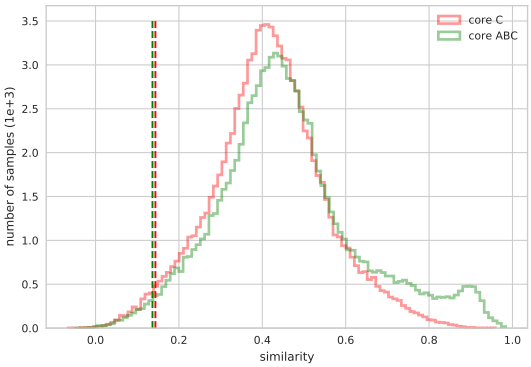

図 2　長単位と内部の中単位の語間の cosine 類似度のヒストグラム．なお縦破線はランダムサンプリングした 10000 件のペア間の cosine 類似度の平均値である (core C = 0.1436, core ABC = 0.1366)．

謝辞

本研究を進めるにあたって，評価コードと再実験結果を共有して頂いた朝日新聞社の田口雄哉氏に謝意を表する．また，ご助言を頂いた奈良先端科学技術大学院大学の松本裕治氏にお礼申し上げる．本研究はワークスアプリケーションズと国立国語研究所の共同研究協定によるものである．

参考文献

[1] 田口雄哉, 田森秀明, 人見雄太, 西鳥羽二郎, 菊田洸：“同義語を考慮した日本語単語分散表現の学習”, 情報処理学会第 233 回自然言語処理研究会.

[2] M. Asahara: “NWJC2Vec: Word embedding dataset from ‘NINJAL Web Japanese Corpus’”, Terminology: International Journal of Theoretical and Applied Issues in Specialized Communication, **24**, 2, pp. 7–25 (2018).

[3] T. Mikolov, K. Chen, G. Corrado and J. Dean: “Efficient estimation of word representations in vector space”, ICLR 2013.

[4] K. Takaoka, S. Hisamoto, N. Kawahara, M. Sakamoto, Y. Uchida and Y. Matsumoto: “Sudachi: a japanese tokenizer for business”, LREC 2018.

[5] M. Asahara, K. Maekawa, M. Imada, S. Kato and H. Konishi: “Archiving and analysing techniques of the ultra-large-scale web-based corpus project of ninjal, japan”, Alexandria, **25**, 1-2, pp. 129–148 (2014).

[6] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov: “Enriching word vectors with subword information”, TACL 2017 (2017).

[7] Y. Sakaizawa and M. Komachi: “Construction of a Japanese Word Similarity Dataset”, LREC 2018.

[8] 猪原敬介, 内海彰：“日本語類似度・関連度データセットの作成”, 言語処理学会第 24 回年次大会.

[9] 橋本泰一, 乾孝司, 村上浩司：“拡張固有表現タグ付きコーパスの構築”, 情報処理学会研究報告自然言語処理 (2008-NL-188).

[10] R. Sennrich, B. Haddow and A. Birch: “Neural machine translation of rare words with subword units”, ACL 2016.