マルチタスク学習を利用した短単位の分散表現から長単位の分散表現の合成

河野 慎司　　古宮 嘉那子
茨城大学工学部情報科
{16t4068t,kanako.komiya.nlp}@vc.ibaraki.ac.jp

1　はじめに

近年、単語の分散表現は自然言語処理における基幹技術となっている。分散表現を取得するにあたって、まず単語の分かち書きをする必要がある。しかし単語境界の違いによっては、単語の分散表現を取得後、単語の分散表現どうしを直接比較ができないときがある。例えば unidic の短単位では「会員」は一単語であるが、「裁判員」は二単語であるため、直接分散表現の比較ができない。そのため、複合語の分散表現を単語の分散表現から合成する手法が必要である。本研究では unidic で利用されている単語単位の「短単位」の分散表現から「長単位」の分散表現の合成を行う。このとき、Komiya ら [2] の係り受けを考慮した長単位合成に関する手法を参考に、短単位の係受け情報の分類タスクを使用し、マルチタスク学習を利用した短単位の分散表現から長単位の分散表現の合成を行った。

2　関連研究

本研究の関連研究として、句の分散表現の生成の研究が挙げられる。Muraoka ら [3] は「形容詞+名詞」や「名詞+名詞」などの係り受け関係に注目し、それぞれの係り受け関係毎に異なる重みで分散表現の合成を行っている。Hashimoto ら [1] は句の意味はその構成要素の単語の意味の組み合わせによって決まると仮定し、句の表現は構成要素の単語の表現から計算できるという「構成的表現」とイディオムとして扱われる「非構成的表現」の両方を同時学習する手法を提案している。Komiya ら [2] は係り受けを 13 種類に定義し、SVM を使用し約 17 万件の短単位と長単位の組み合わせを係り受けごとに分類した。その後、係り受け内で短単位から長単位の合成を行っている。

我々は係り受け情報を用いて単語の分散表現を作成しているが、Shikhar ら [4] は主語・目的語などの構文解析をグラフ畳み込みネットワークで行う SynGCN と、単語の上位・下位関係をもつ SemGCN で単語の分散表現を獲得し、SQuAD タスクに適応している。

3　提案手法

Komiya ら [2] は係り受け分類ごとに長単位の合成を行っているが、我々は一度に長単位の合成を行う。例えば長単位が「講習会」のとき、短単位である「講習」と「会」から長単位の「講習会」の意味を成す分散表現を合成する。

講習 + 会 → 講習会

係り受けを考慮せず長単位の合成を行うとき、短単位のペアをニューラルネットの入力にし、長単位を出力とする。このときのネットワーク図を図 1 に示す。

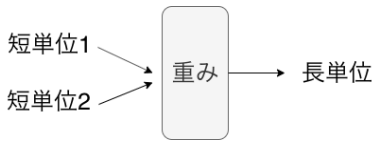

図 1: 短単位のペアから長単位を合成するネットワーク

しかし、この場合だと係り受けを考慮することができない。そこで入力は二つの短単位のまま、係り受けを考慮するため出力に長単位ベクトルの他に、係り受け情報を出力に追加する。このときのネットワーク図を図 2 に示す。このように出力が複数あるネットワークはマルチタスク学習モデルとなる。

人手で係り受け番号を付与したデータ 1,673 件を図 2 のネットワークで学習させる。その後、係り受けが付与されていない二つの短単位と長単位の組み合わせ計 155,779 件でファインチューニング (以下「FT」) を行い、FT をしなかった場合と比較をする。ただし、FT をするときの約 15 万件のデータは係り受けが付与さ

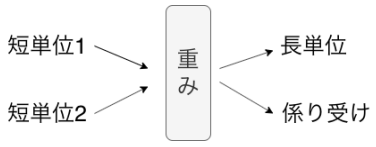

図 2: 係り受けを考慮した長単位合成ネットワーク

れていないため図 3 のように係り受けへの損失 (loss) を 0 とする。

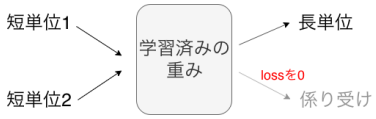

図 3: 係り受けを考慮した長単位合成ネットワーク

また短単位二つと係り受けの 3 入力での長単位合成も行った。まず人手で係り受けを付与したデータで図 2 の構成でマルチタスク学習を行う。次に約 15 万件の短単位を学習済みのモデルに入力し、出力として予測された係り受けを得る。最後に予測係り受けと短単位二つを入力に、長単位を出力として新しくネットワークを学習させ長単位の合成を行う。この一連の流れを図 4 に示す。

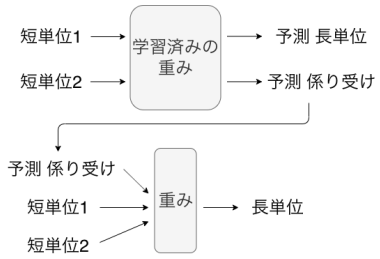

図 4: 予測係り受けと短単位を入力にした長単位合成

4 実験

4.1 短単位・長単位の分散表現の作成

短単位・長単位の分散表現は以下のような手順で作成する。

1. 『現代日本語書き均衡コーパス』の分かち書きを行い、長単位と短単位を一つのファイルとする。
2. 分かち書きのファイルを用いて分散表現を生成する。分散表現の作成には word2vec を利用し、ベクトルの次元数は 100 とした。

短単位の数が 3 単語以上のものは除外した結果、計 195,779 組の短単位のペアと長単位の組み合わせを取得した。また 1,873 件に人手で 13 通りの係り受け番号を付与した。(以下「人手データ」)

4.2 係り受けの定義

本研究で使用する係り受けは Komiya ら [2] と同じ 13 通りとする。以下に 13 通りの係り受けの詳細と具体例を示す。

1. 前の短単位が後の短単位の説明を行う組合せ
   例:「講習会」
2. 目的語と述語の組合せ
   例:「債務放棄」
3. 補語と述語の組合せ
   例:「法的整理」
4. 主語と述語の組合せ
   例:「画面割れ」
5. 一方の短単位がもう一方の短単位の単位となる組合せ
   例:「1 月」
6. 主要単語と接尾語の組合せ
   例:「具体的」
7. 接頭語と主要単語の組合せ
   例:「副代表」
8. 片方の短単位に、助詞が用いられている組合せ
   例:「ための」
9. 固有名詞と一般名詞の組合せ
   例:「茨城県」
10. 名詞と動詞で動詞になる組合せ
    例:「応募する」
11. 数字どうしの組合せ
    例:「三二」
12. 短単位単体では意味を持たず長単位になって初めて意味を持つ組合せ
    例:「だが」
13. その他
    例:「意気揚々」

人手で分類した短単位と長単位の組み合わせは表 1 の通りである。

4.3 ネットワーク構成

二つの短単位の分散表現を入力とし、長単位の分散表現と係り受けを出力するマルチタスク学習を行う。

表 1: 人手で分類した短単位と長単位の組み合わせ

係り受け	1	2	3	4	5	6
件数	447	54	13	12	132	200
7	8	9	10	11	12	13
31	330	97	371	27	32	114

人手データのうち 200 件を開発データとして最適なハイパパラメータを選定した結果、図 5 のネットワーク構成を採用した。共通層である 600 次元の活性化関数には relu、長単位を出力する 100 次元層は sigmoid、係り受けを出力する 13 次元層は softmax を使用した。また最適化アルゴリズムには Adam を選択した。

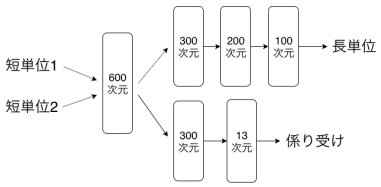

図 5: 学習するネットワーク構成

ただし約 15 万件データで FT を行う際は図 6 のように 600 次元の共通層と係り受けへの層は学習を行わず固定した。

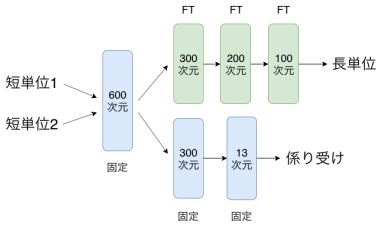

図 6: ファインチューニング (FT) を行う層

短単位二つと予測係り受けを入力とする 3 入力モデルでは中間層一つの 100 次元で実験を行う。

また、約 15 万件データを使って短単位から長単位のみ合成する場合（マルチタスク学習も FT もしない場合)、中間層一つの 300 次元のネットワーク構成で学習した場合が最適であったため、この構成で実験を行う。また、これらの活性化関数には sigmoid を利用した。

4.4 評価方法

取得した分散表現から半数を訓練データ、もう半数をテストデータとして二分割交差検定を行った。この際、生成した長単位の分散表現と正解となる分散表現をコサイン類似度により評価した。

なお学習には Early stopping を導入した。また短単位二つの MaxPooling(最大)、AveragePooling(平均) をベースラインとして設定した。MaxPooling は短単位のペアを同じ次元の位置どうしで比較し値が大きい方を長単位とする手法であり、AveragePooling は短単位のペアを同じ次元の位置どうしで足して 2 で割ることで長単位を生成する手法である。

5 結果

人手データ 1,673 件でマルチタスク学習を行った際の結果を表 2 に示す。

表 2: 係り受けデータのマルチタスク学習性能

長単位のコサイン類似度	係り受け accuracy
0.4951	0.8315

次に約 15 万件の短単位のペアと長単位の組み合わせで係り受けデータで学習された重みを FT した。FT をしない場合（マルチタスク学習も行わない）とベースラインとの比較を表 3 に示す。表 3 での FT あり/なしの () 内は Early stopping での停止エポック数である。

表 3: FT の有無とベースラインの長単位コサイン類似度

FT あり	FT なし	最大	平均
0.5713 (148)	0.5604 (97)	0.3958	0.4448

実験の結果、FT「あり」の方が「なし」に比べ性能がよくなった。

最後に予測係り受けと短単位二つを入力 (3 入力) として長単位の合成を行ったときと、表 3 での FT あり/なしとのコサイン類似度の比較を表 4 に示す。

表 4: 3 入力と FT あり/なしの比較

3 入力コサイン類似度	FT あり	FT なし
0.5626(123)	0.5713 (148)	0.5604 (97)

係り受けを加えた 3 入力のときは FT なしに比べ性能はよくなったが、FT ありに比べると性能は下がる結果となった。

6 考察

本実験の結果により、係り受けを考慮した重みを FT した方が、係り受けを考慮しなかった場合に比べ性能向上につながることが確認できた。この結果は Komiya ら [2] と同様である。

本手法では一度にすべてのデータで学習を行っているが、係り受け分類ごとに長単位の合成性能を調査した。このとき、Komiya ら [2] の研究と比較するため SVM での係り受け分類の性能も求める。本手法で係り受け分類するためには、人手データで学習済みのモデル (図 5) から予測として得られる係り受けをもとに 13 通りの分類を行い、学習を実行した。結果を表 5 に示す。

表 5: 係り受け別のコサイン類似度 () 内は分類件数

係り受け	マルチタスク学習	SVM
1	0.4895 (26,797)	0.4884 (25,198)
2	0.4902 (1,337)	0.4847 (2,196)
3	0.4275 (94)	0.4126 (460)
4	0.4814 (111)	0.5005 (458)
5	0.5572 (11,036)	0.5685 (8,722)
6	0.5409 (20,328)	0.5422 (22,077)
7	0.4957 (1,645)	0.5473 (3,797)
8	0.6820 (49,806)	0.6741 (48,020)
9	0.4239 (5,921)	0.4160 (4,574)
10	0.6024 (24,499)	0.5978 (25,256)
11	0.7425 (261)	0.6009 (802)
12	0.6817 (6,504)	0.6798 (8,017)
13	0.5049 (7,440)	0.4732 (6,202)
マイクロ平均	0.5870	0.5843

分類ごとにみると「係り受け 11」が最も高く、「係り受け 9」が最も低い結果となった。「係り受け 11」が高い性能がでた考察として、「係り受け 11」は数字どうしの組み合わせであるためパターンが少なく短単位・長単位どうしで元々のベクトルが似ているためだと考えられる。一方「係り受け 9」は固有名詞と一般名詞の組み合わせのためパターンが多く、短単位・長単位どうしのベクトルが離れている場合が多いことが要因の一つに挙げられる。

固有名詞と一般名詞の組み合わせには、「人名→一般名詞」や「地名→一般名詞」などのように係り受けをより細かくすることで、約 15 万件データの長単位合成性能も上がるのではないかと考えられる。

7 おわりに

本研究は短単位の分散表現から長単位の分散表現を、係り受けを考慮したマルチタスク学習によって合成する手法を提案した。係り受けを考慮した重みを FT することで、FT をしなかった場合に比べ性能の向上が確認できた。今後の課題としては以下の二点がある。

- 現状と同じ係り受けを利用し、より性能の出るネットワークモデルの構築。
- 係り受けをより細かくする。また係り受けの他に合成に効くサブタスクを追加する。

これらの点を考慮しながら今後研究を進めていきたい。

謝辞

本研究は、茨城大学の若手教員研究費支援制度「分散表現を用いた多単語表現の変換」および JSPS 科研費 18K11421 の助成を受けたものである。また、甲南大学の永田亮先生にさまざまなアドバイスをいただきました。御礼申し上げます。

参考文献

[1] Kazuma Hashimoto and Yoshimasa Tsuruoka. Adaptive joint learning of compositional and non-compositional phrase embeddings. In *Proceedings of the 54th ACL*, pp. 205–215, 2016.

[2] Kanako Komiya, Takumi Seitou, Minoru Sasaki, and Hiroyuki Shinnou. Composing word vectors for japanese compound words using dependency relations. *CICLING*, 2019. no. 229.

[3] Masayasu Muraoka, Sonse Shimaoka, Kazeto Yamamoto, Yotaro Watanabe, Naoaki Okazaki, and Kentaro Inui. Finding The Best Model Among Representative Compositional Models. In *Proceedings of PACLIC 2014*, pp. 65–74, 2014.

[4] Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. *ACL*, p. 3308–3318, 2019.