係り受け関係を用いた句ベクトルの生成

村岡 雅康[†]　島岡 聖世[‡]　山本 風人[†]　渡邉 陽太郎[†]　岡崎 直観[†*]　乾 健太郎[†]
東北大学[†‡]　科学技術振興機構さきがけ[*]
{muraoka, kazeto, yotaro-w, okazaki, inui}@ecei.tohoku.ac.jp[†]
simaokasonse@yahoo.co.jp[‡]

1　はじめに

計算機による自然言語理解に向けて、言葉の意味を正しく計算することは自然言語処理分野において大きな目標の一つである。分布意味論では、単語の意味はそれと同じ文脈で出現する (共起する) 単語によって推定できるというアプローチを取る [9]。例えば、以下のような文を考える。

知人が北海道のお土産に「き花」を買ってきた。

上記の文において、「き花」という単語の意味が分からなかったとしても、それと共起している単語から意味をある程度推測できる。さらに、この未知語が出現する文章を大量に収集すれば、その意味をより精密に推測できるようになる。また、「着物」と「和服」、「ケータイ」と「スマホ」など似た文脈を持つ単語は似た意味を持つと推定できる。このような方法を用いれば、計算機も単語の意味を扱うことができる。

しかし、同様の方法で句や節などを 1 単語とみなし、句や節の意味を捉えようとした時、新たな問題が生じる。それは、句や節が長くなるほどそれらの出現頻度が指数関数的に減少し、正しく意味を推測できなくなるというものである。そこで、句や節、文の意味をそれらを構成する個々の単語の意味から計算するというアプローチ (構成意味論) の研究に注目が集まっている。現在様々な手法が提案されているが、後述するように 3 単語以上の句や文の意味の計算が考慮されていない (再帰性がない)、Socher ら [15] の手法のように特定のタスクを解くことに主眼を置いているなど、句の意味が正しく計算される保証がない。さらに、既存研究では修飾関係 (例:形容詞+名詞) や目的語関係 (例:動詞＋名詞) など明らかに性質の異なる合成に対して、異なる方法で意味の計算を高精度かつ再帰的に行うことができない。これらの問題に対処するため、本稿では係り受け関係に基づくニューラルネットワークモデルを提案する。また、評価実験で提案したモデルの有用性を実験的に示す。

本稿の構成は以下の通りである。まず 2 節では単語ベクトルを構築する方法および単語から構成的に句や文のベクトルを生成する既存のモデルを紹介する。3 節で提案手法を説明し、4 節でモデルの学習方法および評価実験の方法・結果を述べる。最後に 5 節で本研究の全体の総括を行う。

2　関連研究

本節では単語の意味を表すベクトルの代表的な構築方法および構成的に句や文のベクトルを生成する既存研究について述べる。

2.1　単語ベクトルの構築

分布意味論では、単語の意味は d 次元空間上の一点、すなわちベクトルで表す。その構築方法は、以下の 2 種類に大別できる。

・共起頻度を用いる方法 [4, 13] - この方法は各単語についてテキスト中で共起する単語の統計をとり、ノイズ除去・スパースネス解消のため主成分分析 (PCA) 等で次元圧縮して得られたものを単語ベクトルとする方法である。

・ニューラルネットワークを用いた言語モデルで学習する方法 [3, 6] - この方法はランダムに初期化された単語ベクトルを入力とし、言語モデルを学習する過程でニューラルネットの誤差逆伝搬により単語ベクトルを学習する方法である。

2.2　構成的な句ベクトルの生成

まず、構成性の原理および問題の定式化を説明し、その後単語ベクトルから構成的に句や文のベクトルを生成するモデルの代表例を紹介する。

構成性の原理 [10] とは、句や文の意味はそれらに

表 1: 既存モデルの代表例

モデル	数学的表現	パラメータ
add[12, 13]	$w_1\boldsymbol{u} + w_2\boldsymbol{v}$	w_1, w_2
Fulladd[11, 18]	$W_1\boldsymbol{u} + W_2\boldsymbol{v}$	$W_1, W_2 \in \mathbb{R}^{d\times d}$
RNN[17]	$\sigma\left(W\begin{bmatrix}\boldsymbol{u}\\ \boldsymbol{v}\end{bmatrix}\right)$	$W \in \mathbb{R}^{d\times 2d}$
Lexfunc[2, 5]	$A_u\boldsymbol{v}$	$A_u \in \mathbb{R}^{d\times d}$
Fulllex[16]	$\sigma\left([\boldsymbol{W}_1, \boldsymbol{W}_2]\begin{bmatrix}A_u\boldsymbol{v}\\ A_v\boldsymbol{u}\end{bmatrix}\right)$	$W_1, W_2 \in \mathbb{R}^{d\times d}$ $A_u, A_v \in \mathbb{R}^{d\times d}$

σ : 活性化関数 (tanh などのシグモイド関数)

含まれる単語の意味から構成されるという考え方である。例えば、「鮭をくわえた熊の木彫り」という句の意味は、語順を無視すれば、「熊」「鮭」「くわえる」「木彫り」の 4 単語から構成されると考える。句や文の意味を表すベクトルをそれを構成する単語のベクトルから生成することは構成性の原理に基づいている。

以降の説明を容易にするため、「単語ベクトルから句ベクトルを生成する」という問題を数学的に定式化する。2 つの単語ベクトル $\boldsymbol{u}$ と $\boldsymbol{v}$ をあるモデル f に入力として与え、モデル f は句ベクトル $\boldsymbol{p}$ を出力する。これは次のように表せる:

$$\boldsymbol{p} = f(\boldsymbol{u}, \boldsymbol{v}) \tag{1}$$

ただし、関数 f は単語ベクトル $\boldsymbol{u}, \boldsymbol{v}$ から句を表すベクトル $\boldsymbol{p}$ を計算する演算を表す。また、入力と出力のベクトルの次元を等しくすることで再帰的な生成が可能となり、3 単語以上からなる句や文のベクトルを生成できる。

表 1 に、これまでに提案された代表的なモデルを示す。この他に乗算 (**Mult**) モデルや伸張 (**Dil**) モデル [12, 13] などが提案され、Dinu ら [7] はそれらの精度を同一条件下で比較し、Lexfunc モデルが最も優れていると述べている。その理由として、Lexfunc は言語学的な根拠、すなわち単語間に存在する関係 (例えば、修飾-被修飾関係、動詞-目的語関係など) に基づいた合成であることを挙げているが、単語の表現形式を統一していない (形容詞などの従属語を行列、名詞などの主要語をベクトルで表現している) ため、再帰的なベクトルの生成ができないという問題がある。一方、再帰的なベクトルの生成が可能な Fulllex モデルも RNN モデルも非線形変換を行う表現力の高いモデルではあるが、Fulllex モデルは各単語がベクトルと行列で表現されているため学習パラメータが膨大になる。また、RNN モデルは全ての単語の合成を一つの重み行列で行っているため、全ての単語の組み合わせに対応できるベクトルの合成を行うには自由度が不足し、Lexfunc に劣っている。以上をまとめると表 2 のようになる。

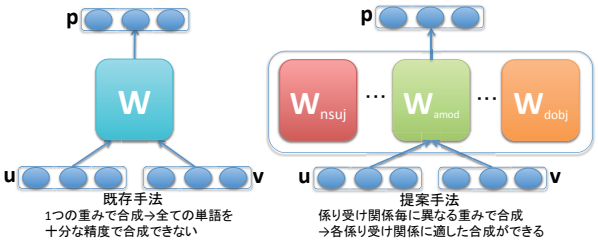

図 1: 既存手法と提案手法の比較

表 2: Lexfunc と Fulllex,RNN および提案手法の違い

	再帰性	表現力	言語学的根拠
Lexfunc[2, 5]	×	○	○
Fulllex[16]	○	△	×
RNN[17]	○	△	×
提案手法	○	○	○

従って、本稿では Fulllex や RNN と同様に再帰的な生成が可能かつ、パラメータ数が Fulllex と RNN のおよそ中間で、Lexfunc と同様に言語学的根拠に基づいた合成を行うモデルを提案する。具体的には合成時に使用する重み行列を係り受け関係によって使い分けるニューラルネットワークモデルを提案する。係り受け関係に着目したのは以下の理由による。本研究に類似したモデルとして、Socher ら [15] が提案した品詞毎に重みを使い分けるモデルがある。パラメータ数は提案手法と同様に Fulllex と RNN のおよそ中間であるものの、このモデルは正しい構文木を構築するという目的で設計され、合成されたベクトルには合成前の要素の統語情報が含まれていれば十分その目的は達成される。つまり、その合成が自然かどうか (例えば、... 前置詞 限定詞 名詞 ... のような単語列に対して前置詞と限定詞の合成ではなく、限定詞と名詞の合成が正しいこと) を合成されたベクトルから識別し、その合成を行うかどうかを決定できればよい。しかし、その合成されたベクトルが句や文の意味を正しく表現できているとは限らない。そこで統語的な意味役割によって分類された係り受け関係を用いることで、この問題の解消が期待できると考えた。

3 提案手法

一般的なニューラルネットによる句ベクトルの生成は次式で表される (図 1 左):

$$\boldsymbol{p} = f(\boldsymbol{u}, \boldsymbol{v}) = \sigma\left(W\begin{bmatrix}\boldsymbol{u}\\ \boldsymbol{v}\\ b\end{bmatrix}\right) \tag{2}$$

ただし、$\boldsymbol{u}, \boldsymbol{v}$ は d 次元列ベクトル、W は $d \times (2d+1)$ 行列、b はバイアス項 (スカラ) である。また、$\sigma(.)$ はシグモイド関数であり要素毎に適用する。本研究では tanh を用いた。このモデルでは、形容詞による名詞の修飾や主語と動詞の合成、動詞と目的語の合成等、明らかに性質の異なる合成を 1 つの重みで行うため、それらすべてを十分な精度で合成することは困難である。これに対し提案手法は、2 つの単語ベクトルに加えてそれらの間に成り立つ係り受け関係 r も入力として使用する。この関係 r によって重み行列 W_r を決定する (図 1 右):

$$\boldsymbol{p} = f(\boldsymbol{u}, \boldsymbol{v}, r) = \sigma\left(W_r \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ b_r \end{bmatrix}\right) \tag{3}$$

ここで $W_r \in \mathbb{R}^{d \times (2d+1)}, b_r \in \mathbb{R}$ は係り受け関係の種類だけ用意する。これにより、係り受け関係毎に異なる性質の合成を異なる重み行列で学習するため、既存のモデルでは達成できなかったより精密な合成が可能となる。

本研究の目的は、入力となる単語ベクトル $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{d \times 1}$ が与えられたとき、それらによって構成される句の意味を正しく表すベクトル $\boldsymbol{p} \in \mathbb{R}^{d \times 1}$ を出力するための関数 $f : \mathbb{R}^{(2d+1) \times 1} \longrightarrow \mathbb{R}^{d \times 1}$、すなわち重み行列 $\{W_r \in \mathbb{R}^{d \times (2d+1)}\}$ を学習することである。これは次のような最適化問題を解くことで実現される。
訓練データ集合 $\{((\boldsymbol{u}_i, \boldsymbol{v}_i), \boldsymbol{t}_i) | i = 1...N\}$($\boldsymbol{t}_i$: 教師句ベクトル) に対して定義される以下の誤差関数 $J(\theta)$ を最小化する:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} ||\boldsymbol{p}_i - \boldsymbol{t}_i||^2 + \frac{\lambda}{2} ||\theta||^2 \tag{4}$$

ただし、θ は学習パラメータの全てを表し、$\theta = \langle W_r, b_r | r$: 係り受け関係 $\rangle$、λ は L2 正則化項のパラメータである。勾配の計算は一般的なニューラルネットワークモデルと同様に誤差逆伝播法を用いる。

4 評価実験

本節では、まず入力および教師ベクトルとなる単語・句ベクトルの作成方法およびモデルの学習方法を説明し、その後学習されたモデルの評価実験について述べる。

4.1 単語・句ベクトルの作成

単語・句ベクトルは Dinu らの論文 [7] を参考とし、以下のような手順で作成する。

1. ukWaC[1]†,Wikipedia(2009)[1]†,ClueWeb09‡の計約 38 億トークンからなるコーパスから内容語 (名詞、形容詞、動詞、副詞) の頻度統計を求め、その結果の名詞・形容詞・動詞のみからなる上位 1 万語を語彙 V とする。
2. 語彙 V に含まれる単語のみで構成される 2 単語の句 (形容詞+名詞、名詞+名詞、動詞+名詞) の頻度を求める。
3. 語彙 V に含まれる各単語および出現頻度 1000 以上の句のそれぞれに対し、コーパス中で同一文内かつ前後 50 語以内に共起する内容語の頻度統計を求め、単語・句と文脈語の共起行列を作る。
4. 共起行列の各要素の値を PMI(相互情報量)[8] に変換する。
5. 共起行列を EM アルゴリズムを用いた PCA[14][1]で $d = 200$ 次元に圧縮する。

これにより 10,000 種類の単語ベクトルおよび 17,433 種類の句ベクトルが得られた。

4.2 定量評価

評価実験は Mitchell ら [13][2]によって作成されたデータセットを用いて行った。

このデータセットは $\langle$ 句 1, 句 2, 類似度 $\rangle$ の 3 つ組を 1 事例とし、2 つの句の意味的類似度を 7 段階で人手で付与してある。3 種類の品詞の組み合わせ (形容詞+名詞, 名詞+名詞, 動詞+名詞) 毎に 108 事例ある。例えば、vast amount と large quantity は類似度 7、hear word と remember name は類似度 1、といった正解事例が収められている。評価は、提案手法が生成した句ベクトルのコサイン類似度と正解データの類似度とのスピアマン順位相関係数を計算し、モデルが生成した句ベクトルの類似度が人手による判断とどれだけ近くなるかを測定する。相関係数が高いほど、人間に近い類似性判断ができたことになり、モデルの性能が高いことを意味する。本研究では、Mitchell ら [13] と同様に、人手の類似度は平均値ではなく、それぞれ別のデータ点として計算した。

4.3 モデルの学習

評価データに含まれる句のベクトルを除いた 16,845 種類の句を、それぞれコーパス中の出現回数の 1000 分の 1 回だけ重複して出現させた合計 $N = 175,899$

†http://wacky.sslmit.unibo.it/
‡http://lemurproject.org/clueweb09/
[1]大規模データに対応するためオンラインアルゴリズムに拡張した。
[2]http://homepages.inf.ed.ac.uk/s0453356/share

句を訓練データとした。その他のハイパーパラメータは次のように設定した。

・学習率 $\alpha = 1.0 \times 1.1^{-l}$($l$: 現在の反復回数)

・L2 正則化項 $\lambda = 0.1$

・反復回数の上限: 100

重み行列は 31 種類の係り受け関係に対して定義し、それぞれ学習開始時に次のように初期化した。

$$W_r = 0.01[I_{n\times n} I_{n\times n} 0_{n\times 1}] + \mathcal{N}(0_{(2d+1)\times 1}, 0.001 I_{(2d+1)\times d}) \quad (5)$$

学習は 2.2GHz の計算機サーバ上で 10 並列で行い、学習に要した時間は約 7 時間であった[3]。

4.4 結果・考察

表 3 に評価結果を示す。corpus は句の共起ベクトル、add は表 1 において $w_1 = w_2 = 1.0$ とした加算モデルである。upper-bound は被験者間の相関であり、被験者 1 人とその他の被験者の相関を求め、最後にそれらの平均を取ることで算出した。corpus の精度が低いのは、そもそも評価データに含まれる句がコーパス中に出現せず、句ベクトル自体が求まっていないか (その場合の類似度は 0 とした)、出現頻度が小さすぎて十分な情報量を持ったベクトルにならなかったためと考えられる。また、全ての相関係数の間には統計的に有意な差が認められた ($p < 0.01$)。評価結果では add モデルが RNN や Fulladd よりも優れた強いベースラインになっているが、全ての句カテゴリにおいて提案手法が add モデルの精度を上回ったことが確認できた。このことより、句ベクトルの生成において係り受け関係毎に重み行列を個別に学習することは有用であることが確かめられた。

5 おわりに

本稿では、ニューラルネットを用いた句ベクトルの生成に注目し、係り受け関係毎に異なる重み行列を用いることの有用性について調査を行った。その結果、既存手法を上回り、本手法の有用性を確認できた。今後は、提案手法を拡張し、再帰的合成を行った場合の精度を言い換えのタスク等を通じて調査・検討したい。

謝辞

本研究は、文部科学省科研費 (23240018) および JST 戦略的創造研究推進事業「さきがけ」から部分的な支援を受けて行われた。

[3]Python の numpy や multiprocessing のモジュールを使用した。

表 3: カテゴリ別スピアマン相関係数

	形容詞+名詞	名詞+名詞	動詞+名詞
corpus	0.362	0.432	0.215
add[12, 13]	0.442	0.432	0.404
Fulladd[11, 18]	0.406	0.420	0.366
RNN[17]	0.424	0.416	0.379
提案手法	**0.450**	**0.464**	**0.411**
upper-bound	0.539	0.490	0.505

全ての相関係数の間には統計的に有意な差がある ($p < 0.01$)

参考文献

[1] M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. In *Language Resources and Evaluation*, Vol. 43, pp. 209–226, 2009.

[2] M. Baroni and R. Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *EMNLP*, pp. 1183–1193, 2010.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, Vol. 3, pp. 1137–1155, 2003.

[4] J. Bullinaria and J. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. In *Behavior Research Methods*, pp. 510–526.

[5] B. Coecke, M. Sadrzadeh, and S. Clark. Mathematical foundations for a compositional distributional model of meaning. In *Linguistic Analysis*, No. 36, pp. 345–384, 2010.

[6] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008.

[7] G. Dinu, N. T. Pham, and M.Baroni. General estimation and evaluation of compositional distributional semantics models. In *the Workshop on Continuous Vector Space Models and their Compositionality*, 2013.

[8] S. Evert. The statistics of word cooccurrences. *Dissertation*, 2005.

[9] J. R. Firth. *Papers in linguistics 1934-51*. Oxford University Press, 1957.

[10] G. Frege. On sense and reference. In *Ludlow (1997)*, pp. 563–584, 1892.

[11] E. Guevara. A regression model of adjective-noun compositionality in distributional semantics. In *GEMS*, pp. 33–37, 2010.

[12] J. Mitchell and M. Lapata. Vector-based models of semantic composition. In *ACL*, pp. 236–244, 2008.

[13] J. Mitchell and M. Lapata. Composition in distributional models of semantics. In *Cognitive Science*, pp. 1388–1429, 2010.

[14] S. Roweis. EM Algorithms for PCA and SPCA. In *Neural Information Systems 10 (NIPS'97)*, pp. 626–632, 1998.

[15] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *ACL*, 2013.

[16] R. Socher, B. huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP*, pp. 1201–1211, 2012.

[17] R. Socher, C. Lin, A. Y. Ng, and C. D. Manning. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, 2011.

[18] F. Zanzotto, I. Korkontzelos, F. Falucchi, and S. Manandhar. Estimating linear models for compositional distributional semantics. In *COLING*, pp. 1263–1271, 2010.