特　集「ニューラルネットワーク研究のフロンティア」

言語処理における分散表現学習のフロンティア

Frontiers in Distributed Representations for Natural Language Processing

岡崎　直観　　東北大学大学院情報科学研究科
Naoaki Okazaki　　Graduate Schoolof Information Sciences,Tohoku University.
okazaki@ecei.tohoku.ac.jp, http://www.chokkan.org/

Keywords: distributed representation, neural network, natural language processing.

1. は　じ　め　に

言語の意味をコンピュータ上でどのように表現すればよいか？　これは言語処理の長年の未解決問題である．曖昧性，多様性，構成性，推論など，言語が関与するさまざまな現象を統一的に扱う理論と，言語の意味表現は表裏一体の課題であり，すべてを同時に解決するのは難しい．

ところが，最近ニューラルネットワーク（Neural Network：NN）で言語処理のタスクを一貫してモデル化する研究が急増している [Goldberg 15, 坪井 15]．これらの研究は，**分散表現**（distributed representation）[Hinton 86] に基づき，単語や文を NN 上で表現している*1．**局所表現**（local representation）では各概念に一つのニューロン（計算ユニット）を割り当てる．これに対し，分散表現では，各概念は複数のニューロンの活性パターンで表現され，各ニューロンも，複数の概念から参照される（図 1）．通常，単語や文の意味は固定数のニューロンの活性パターンで表現される．ニューロンの活性値を実数値ベクトルで表せば，分散表現は単語や文の意味を**ベクトル空間**（vector space）で表現していることになる．

拍子抜けするほど単純な表現形式であるが，品詞タグ付け [Tsuboi 14]，文書分類 [Le 14]，評判分析 [Kim 14, Socher 13]，機械翻訳 [Cho 14, Sutskever 14] など，さまざまな言語処理タスクで分散表現が成功を収めている．これらの研究の多くは，単語（または文字）の分散表現から句や文の分散表現を合成し，応用タスクを解いている．そこで，本稿は分散表現に関する最新の研究動向を，単語の分散表現（2 章），および句や文の分散表現（3 章）という章立てで解説する．

なお，本稿の読者は，ソフトマックス関数，活性化関数，確率的勾配降下法，誤差逆伝播法など，NN の基礎を一通り理解していると想定する．本稿で用いる記法は，元論文の忠実な再現ではなく，各章内で統一されたものになるように工夫した．シグモイド関数 σ および tanh がベクトルを引数に取る場合は，要素ごとに計算した結果をベクトルで返すこととする．

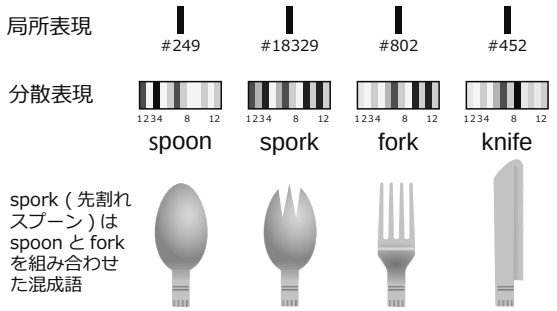

図 1　局所表現と分散表現の例．
局所表現では spoon, fork, spork にそれぞれ，249 番，802 番，18 329 番など，別々のニューロンが割り当てられる．分散表現では，各概念は複数のニューロンの活性パターンで表現され，各ニューロンも複数の概念から参照される．この図では，各概念を 12 個のニューロンの活性状態（濃淡）で表現している．spork の活性パターンを spoon と fork の活性パターンの組み合わせたようなものにすれば，spork が spoon と fork の特徴（食べる，すくう，刺す，金属など）を併せもつことを表現できる

2. 単語の分散表現

2·1　単語文脈行列

単語の意味を統計的に学習する手法は，**分布仮説** [Harris 54] をおくことが多い．この仮説は，*You shall know a word by the company it keeps* という有名な格言 [Firth 57] が示唆するように，単語をその文脈の出現分布で表したとき，文脈の分布の類似性と単語の意味の類似性には相関があることを表現している．

分布仮説を定式化する．コーパス中の各単語について，その周辺（例えば前後 δ 語）に出現する単語を**文脈語**あるいは**文脈**と呼ぶ．コーパス中に出現するすべての単語の集合を V，すべての文脈の集合を C とする．単語 i と

*1 単語や文などの入力を NN の空間で表現することを埋め込む（embed）という．単語埋込み（word embeddings）とは，単語をニューロンの活性パターンに対応付けるものを指す．

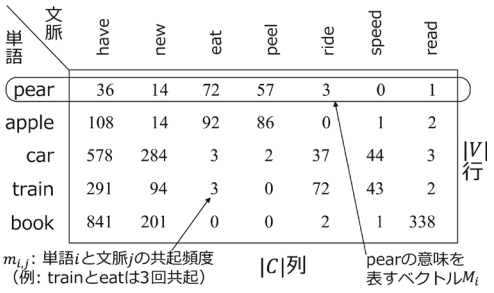

図2 共起頻度を要素とする単語文脈行列の例

文脈jの共起の強さを行列$M\in\mathbb{R}^{|V|\times|C|}$で表現したものを，単語文脈行列と呼ぶ．単語文脈行列Mの定義はさまざまである．最も単純なのは，単語iの周辺に文脈jが出現した回数$\#(i,j)$（単語iと文脈jの共起回数）を行列Mの各要素$m_{i,j}$とすることである（図2）．

$$m_{i,j}^{\mathrm{FREQ}} \equiv \#(i,j) \tag{1}$$

他にも，さまざまな共起尺度が提案・検討されてきた．例えば，正の自己相互情報量（PPMI）[Bullinaria 07]は，頻出文脈（例えば*have*や*new*など）の影響を軽減する．

$$m_{i,j}^{\mathrm{PPMI}} \equiv \max\{0,\mathrm{PMI}(i,j)\} \tag{2}$$

$$= \max\left(0,\log\frac{\#(*,*)\times\#(i,j)}{\#(i,*)\times\#(*,j)}\right) \tag{3}$$

ここで，$\#(i,*)=\sum_j\#(i,j)$，$\#(*,j)=\sum_i\#(i,j)$，$\#(*,*)=\sum_{i,j}\#(i,j)$である．

単語文脈行列Mの行ベクトルM_iを，単語iの**単語ベクトル**と呼ぶ．ベクトル空間法 [Salton 75, Turney 10]で文書とクエリの類似度をモデル化したのと同様に，単語iと単語kの類似度は，ベクトルM_iとM_kのなす角θの余弦（**コサイン類似度**）として求めることができる．

$$\mathrm{sim}(i,k)=\cos\theta=\cos(M_i,M_k)=\frac{M_i\cdot M_k}{|M_i||M_k|} \tag{4}$$

2·2 潜在的意味解析

前節で説明した方法で求めた単語ベクトルの次元数は$|C|$であるため，分散表現としてNNに埋め込むには大量のニューロンが必要になる．また，「食べ物」や「食料」など，同じ意味をもつ文脈が別々の次元をとるため，これらの共通性を活用できない．そこで，単語文脈行列の情報をできるだけ保持しつつ，低次元密行列に圧縮し，NNに埋め込みやすい分散表現に変換したい．

ここでは，**潜在的意味解析**（Latent Semantic Analysis：LSA）[Deerwester 90]を説明する．LSAは，行列Mを**特異値分解**（Singular Value Decomposition：SVD）で直交行列U，対角行列Σ，直交行列V^{T}の積に分解する*2．

$$\overbrace{M}^{(m\times n)} = \overbrace{U}^{(m\times m)}\cdot\overbrace{\Sigma}^{(m\times n)}\cdot\overbrace{V^{\mathsf{T}}}^{(n\times n)} \tag{5}$$

対角行列Σは行列の特異値Mを対角要素に並べたもので，$r=\mathrm{rank}(M)$とすると，次式の形式をとる．

$$\Sigma=\mathrm{diag}(\sigma_1,...,\sigma_r,\sigma_{r+1},...,\sigma_{\min\{m,n\}}),$$
$$\sigma_1\geq...\geq\sigma_r>\sigma_{r+1}=...=\sigma_{\min\{m,n\}}=0 \tag{6}$$

ここで，対角行列Σの代わりに，d個（$d<r$）の特異値で打ち切った行列$\tilde{\Sigma}$を考える．

$$\tilde{\Sigma}=\mathrm{diag}(\sigma_1,...,\sigma_d,\sigma_{d+1},...,\sigma_{\min\{m,n\}}),$$
$$\sigma_1\geq...\geq\sigma_d>\sigma_{d+1}=...=\sigma_{\min\{m,n\}}=0 \tag{7}$$

この対角行列$\tilde{\Sigma}$を用い，$\tilde{M}=U\cdot\tilde{\Sigma}\cdot V^{\mathsf{T}}$を再構成したものは，行列$M$の$d$ランク近似と呼ばれる．この行列$M$は，階数$d$の行列の中で，誤差$\|M-\tilde{M}\|_2$が最小のものであることが知られている*3．

SVDの結果より，行列Mを（$|V|\times d$）の行列に圧縮した行列M'は次式で与えられる．

$$M'=U\cdot\tilde{\Sigma}^{\gamma} \tag{8}$$

M'は（$|V|\times|C|$）行列であるが，d個（$d<r$）の特異値で打ち切った対角行列$\tilde{\Sigma}$を用いたので，$d+1$列目以降はすべて0になり，実質的に（$|V|\times d$）行列とみなせる．なお，SVDは式（5）の完全な形ではなく，dランク近似を効率良く求める手法 [Halko 11]が用いられる．

乗数パラメータγの値として，0, 0.5, 1が有用である．$\gamma=0$は，$\tilde{\Sigma}$の特異値をすべて1とみなし，特異値を無視することに相当する．$\gamma=1$の場合は，次元圧縮後の行列M'が，圧縮前の行列Mの単語ベクトルの類似度を保存する*4．$\gamma=0.5$は，特異値の情報をUとV^{T}に等しく分配すると解釈できる．

2·3 ニューラル確率言語モデル

2010年頃に深層学習がブレイクする前から，言語処理のタスクをNNでモデル化し，その学習（誤差逆伝播法）の副産物として単語の分散表現を獲得する研究が行われていた．その先駆けと位置付けられる研究が，**ニューラル確率言語モデル** [Bengio 03]である．

学習コーパスをT個の単語列$w_1, w_2, \cdots, w_T$で表し，各単語w_tは集合Vの要素であるとする．確率的言語モデルでは，単語列の確率を計算するため，過去の単語の出現履歴$w_1, \cdots, w_{t-1}$から次の単語w_tの出現を予測するモデルを学習することが多い．[Bengio 03]は，直前のδ個の単語列$w_{t-\delta}, \cdots, w_{t-1}$から位置$t$の単語$w_t$を予測する確率分布を，ソフトマックス関数でモデル化した．

$$p(w_t|w_{t-1},...,w_{t-\delta})=\frac{\exp(y_{w_t})}{\sum_{w'\in V}\exp(y_{w'})} \tag{9}$$

*2 単語の集合Vと記法が衝突しているが，SVDでは行列U, Σ, V^{T}の積で表現する慣習なので，記法の衝突の回避を諦めた．ここでは，V^{T}を行列と解釈していただきたい．

*3 $\|A\|_2$は行列Aのフロベニウスノルムである．

*4 $\tilde{M}\tilde{M}^{\mathsf{T}}=U\tilde{\Sigma}V^{\mathsf{T}}V\tilde{\Sigma}U^{\mathsf{T}}=(U\tilde{\Sigma})(U\tilde{\Sigma})^{\mathsf{T}}$であることによる．

ここで，y_w は単語 w の出現を予測するスコアで，$|V|$ 個の単語に関して計算される．このスコアを $|V|$ 次元ベクトル $\boldsymbol{y}=(y_1,\cdots,y_{|V|})^\mathsf{T}$ で表現したものを NN で求める．

$$\boldsymbol{y}=\boldsymbol{b}+W\boldsymbol{x}+U\tanh(\boldsymbol{d}+H\boldsymbol{x}) \tag{10}$$

$$\boldsymbol{x}=\boldsymbol{v}_{w_{t-1}}\oplus\boldsymbol{v}_{w_{t-2}}\oplus\ldots\oplus\boldsymbol{v}_{w_{t-\delta}} \tag{11}$$

ここで，$\boldsymbol{v}_w\in\mathbb{R}^d$ は単語 w の分散表現，$\oplus$ はベクトルを連結する演算子である．ゆえに，ベクトル $\boldsymbol{x}$ は過去の単語列 $w_{t-\delta},\cdots,w_{t-1}$ の分散表現を連結し，$d\delta$ 次元ベクトルで表現したものである．また，隠れ層の次元をパラメータ h で設定すると，$H\in\mathbb{R}^{h\times d\delta}$，$\boldsymbol{d}\in\mathbb{R}^h$，$U\in\mathbb{R}^{|V|\times h}$，$W\in\mathbb{R}^{|V|\times d\delta}$，$\boldsymbol{b}\in\mathbb{R}^{|V|}$である．式（9）から定義される対数尤度を最大化するように確率的勾配降下法および誤差逆伝播法を適用し，単語の分散表現 $\boldsymbol{v}_w$，行列 H, U, W, バイアス項 $\boldsymbol{b},\boldsymbol{d}$ を学習する．

この研究の後，式（9）の代わりに**双対数線形モデル**（log-bilinear model）を導入した研究 [Mnih 07] や，式（9）のソフトマックスの計算を階層的に構成し，計算量を削減する研究 [Mnih 09, Morin 05] などが発表されている．

2·4 SENNA

[Collobert 08, Collobert 11] は，単一の**畳込みニューラルネットワーク**（Convolutional Neural Network：CNN）で，品詞タグ付け，チャンキング，固有表現抽出，意味役割付与などの複数の言語処理タスクを同時に解く枠組み（SENNA）を提案した．言語処理への CNN の適用やマルチタスク学習など，先駆者的な研究であるが，NN による単語の分散表現の学習においても，新しい提案を残している．それは，前に出現した単語列から次の単語を予測する言語モデルではなく，前後に出現した単語列から中央の単語を予測するモデルを学習するほうが，単語の分散表現を効率良く獲得できるというアイディアである．

このアイディアは以下のように定式化される*5．学習コーパス中のある単語 w_t に関して，その前後 δ 語を文脈と考える．コーパス中に出現した単語列 $w_{t-\delta},\cdots,w_t,\cdots,w_{t+\delta}$ と，中央の単語 w_t を適当な単語 $w'\in V$ に置換した単語列 $w_{t-\delta},\cdots,w',\cdots,w_{t+\delta}$ を識別できるような NN を学習するため，以下の目的関数を最小化する．

$$\sum_{t=1}^{T}\sum_{w'\in V}\max\bigl(0,1-f(\boldsymbol{x}_t)+f(\boldsymbol{x}_t^{w'})\bigr) \tag{12}$$

ただし，$f(\boldsymbol{x})$ は行列 H で $(2\delta+1)d$ 次元のベクトル $\boldsymbol{x}$ を h 次元の隠れ層に写像し，活性化関数（tanh）を経た後，ベクトル $\boldsymbol{b}$ との内積でスコアを出力する NN である．

$$f(\boldsymbol{x})=\boldsymbol{b}\cdot\tanh(H\boldsymbol{x}) \tag{13}$$

また，$\boldsymbol{x}_t$ と $\boldsymbol{x}_t^{w'}$ はそれぞれ，単語列 $w_{t-\delta},\cdots,w_t,\cdots,w_{t+\delta}$ の分散表現の連結，および中央の単語 w_t を w' に置換した場合の分散表現の連結である．

$$\boldsymbol{x}_t=\boldsymbol{v}_{w_{t-\delta}}\oplus\ldots\oplus\boldsymbol{v}_{w_t}\oplus\ldots\oplus\boldsymbol{v}_{w_{t+\delta}} \tag{14}$$

$$\boldsymbol{x}_t^{w'}=\boldsymbol{v}_{w_{t-\delta}}\oplus\ldots\oplus\boldsymbol{v}_{w'}\oplus\ldots\oplus\boldsymbol{v}_{w_{t+\delta}} \tag{15}$$

2·5 word2vec と類推

最近の分散表現研究ブームの火付け役になったのが，word2vec*6[Mikolov 13] である．この手法は，SENNA のアイディアを双対数線形モデルに適用し，負例サンプリングや階層的ソフトマックスなどで学習を高速化したものと見ることもできる．本稿では，負例サンプリングに基づく手法を解説する．

学習コーパスを単語列 $w_1, w_2, \cdots, w_T$ で表す．ある位置 t で出現する単語 w_t に対して，その前後 δ 個の単語列を文脈窓 $C_{w_t}=(w_{t-\delta},\cdots,w_{t-1},w_{t+1},\cdots,w_{t+\delta})$ とする．Mikolov らの手法は，文脈窓 C_{w_t} から単語 w_t を予測する条件付き確率分布関数 $p(w_t|C_{w_t})$ を定義し*7，その対数尤度 $\mathcal{L}$ を最大化する．

$$\mathcal{L}\equiv\sum_{t=1}^{T}\log p(w_t|C_{w_t}) \tag{16}$$

$p(w_t|C_{w_t})$ は，skip-gram（SG）もしくは continuous bag-of-words（CBOW）でモデル化される．

SG では，$p(w_t|C_{w_t})$ を各文脈語 $c\in C_{w_t}$ から単語 w_t を予測する条件付き確率 $p(w_t|c)$ の積に分解する．

$$\mathcal{L}^{\mathrm{SG}}\equiv\sum_{t=1}^{T}\sum_{c\in C_{w_t}}\log p(w_t|c) \tag{17}$$

さらに，単語 c から単語 w_t を予測する条件付き確率 $p(w_t|c)$ を双対数線形モデルで定式化する．

$$p(w_t|c)\equiv\frac{\exp(\boldsymbol{v}_c\cdot\tilde{\boldsymbol{v}}_{w_t})}{\sum_{w'\in V}\exp(\boldsymbol{v}_c\cdot\tilde{\boldsymbol{v}}_{w'})} \tag{18}$$

ここで，$\boldsymbol{v}_c$ は単語 c のベクトル，$\tilde{\boldsymbol{v}}_w$ は単語 w を予測するベクトルである*8．式（18）の分母の計算には，内積と exp の計算が $|V|$ 回必要となり，大規模な学習データに対応できない．そこで，V に関する多値分類問題を，

---

*5 元論文 [Collobert 08, Collobert 11] は NN の詳細を割愛しているため，[Bengio 09] および [Turian 10] に基づいて説明する．

*6 https://code.google.com/p/word2vec/

*7 元論文 [Mikolov 13] において，Skip-gram モデルは単語 w から文脈 c を予測する条件付き確率分布 $p(c|w)$ をモデル化すると説明されているが，word2vec の実装が実際にモデル化しているのは $p(w|c)$ である．本稿では，word2vec の実装に忠実にモデルを説明する．word2vec では単語と文脈に対称性がある（t が動くと単語と文脈語の役割が入れ替わる）ため，どちらの向きでも同様の学習をしていることになる．

*8 $\boldsymbol{v}$ と $\tilde{\boldsymbol{v}}$ の 2 系統のベクトルがある．word2vec が単語ベクトルとして保存するのは $\forall c\in C:\boldsymbol{v}_c$ である．表記上は文脈語 c から単語ベクトルを取っているように見えるため，違和感を感じるかもしれない．word2vec の実装では単語と文脈語の集合は等しいため（$V=C$），$\boldsymbol{v}_c$ を単語 c のベクトルとみなすことができる．

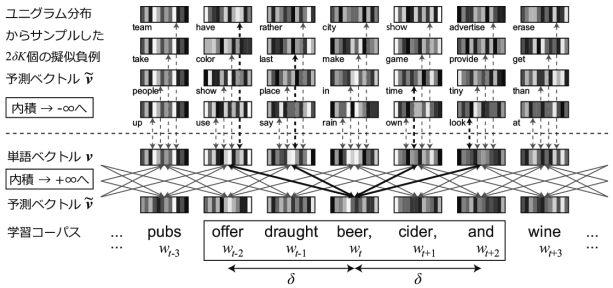

図3 負例サンプリングに基づく skip-gram（SGNS）の動作例（文脈幅 $\delta=2$，擬似負例数 $K=1$ とした）．
実線矢印は内積値が $+\infty$，点線矢印は内積値が $-\infty$ になるようにベクトルを更新する．例えば，w_t = *beer*, 文脈語 w_{t-2} = *offer* に関して擬似負例単語 $\breve{w}_t$ = *have* をサンプルしたとき，$\boldsymbol{v}_{\text{offer}} \cdot \tilde{\boldsymbol{v}}_{\text{beer}} \to +\infty$，$\boldsymbol{v}_{\text{offer}} \cdot \tilde{\boldsymbol{v}}_{\text{have}} \to -\infty$ を目標に，$\boldsymbol{v}_{\text{offer}}$, $\tilde{\boldsymbol{v}}_{\text{beer}}$, $\tilde{\boldsymbol{v}}_{\text{have}}$ を更新する（確率的勾配降下法）．同様の処理を文脈語 *draught*, *cider*, *and* に関して実行すると，w_t = *beer* に対する処理が完了する．この一連の処理を学習コーパスの先頭の単語 w_1 から末尾の単語 w_T まで繰り返す

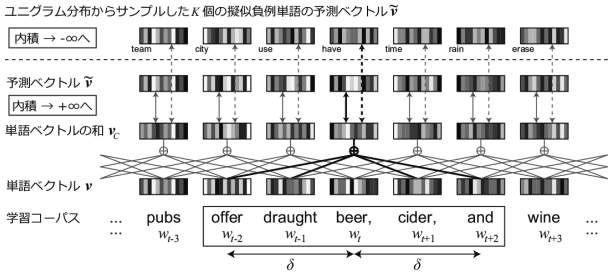

図4 負例サンプリングに基づく continuous bag-of-words の動作例（文脈幅 $\delta=2$，擬似負例数 $K=1$ とした）．
実線矢印は内積値が $+\infty$，点線矢印は内積値が $-\infty$ になるようにベクトルを更新する．例えば，w_t = *beer* のとき，*offer, draught, cider, and* の単語ベクトルの和 $\boldsymbol{v}_C$ を計算する．擬似負例単語 $\breve{w}_t$ = *have* をサンプルしたとき，$\boldsymbol{v}_C \cdot \tilde{\boldsymbol{v}}_{\text{beer}} \to +\infty$，$\boldsymbol{v}_C \cdot \tilde{\boldsymbol{v}}_{\text{have}} \to -\infty$ を目標に，$\boldsymbol{v}_{\text{offer}}$, $\boldsymbol{v}_{\text{draught}}$, $\boldsymbol{v}_{\text{cider}}$, $\boldsymbol{v}_{\text{and}}$, $\tilde{\boldsymbol{v}}_{\text{beer}}$, $\tilde{\boldsymbol{v}}_{\text{have}}$ を更新する（確率的勾配降下法と誤差逆伝播法）．この処理を学習コーパスの先頭の単語 w_1 から末尾の単語 w_T まで繰り返す

$K+1$ 回の二値分類（ロジスティック回帰）で近似する．

$$\mathcal{L}^{\text{SG}} = \sum_{t=1}^{T} \sum_{c \in C_{w_t}} \left(\log \sigma(\boldsymbol{v}_c \cdot \tilde{\boldsymbol{v}}_{w_t}) + \sum_{k=1}^{K} \log \sigma(-\boldsymbol{v}_c \cdot \tilde{\boldsymbol{v}}_{\breve{w}'}) \right) \tag{19}$$

ここで，σ はシグモイド関数，$\breve{w}'$ は学習データの単語のユニグラム分布*9からランダムにサンプルした擬似負例単語（ただし，Σ_k の反復ごとに K 回サンプリングする）である．式（19）は，学習データに出現する単語・文脈ペア $\langle w_t, c \rangle$ に対して 1，単語をランダムに置換（$w_t \to \breve{w}'$）したペア $\langle \breve{w}', c \rangle$ に対して 0 を予測するロジスティック回帰モデルを学習する解釈するとわかりやすい（図3）．以降，負例サンプリングに基づく SG を SGNS と略す．

一方，CBOW では $p(w_t | C_{w_t})$ の予測を確率の積に分解せず，文脈語のベクトルの和 $\boldsymbol{v}_{C_t} = \sum_{c \in C_{w_t}} \boldsymbol{v}_c$ を用いて直接モデル化する（図4）．その他は SG と同様である．

$$\mathcal{L}^{\text{CBOW}} = \sum_{t=1}^{T} \left(\log \sigma(\boldsymbol{v}_{C_t} \cdot \tilde{\boldsymbol{v}}_{w_t}) + \sum_{k=1}^{K} \log \sigma(-\boldsymbol{v}_{C_t} \cdot \tilde{\boldsymbol{v}}_{\breve{w}'}) \right) \tag{20}$$

$$\boldsymbol{v}_{C_t} = \sum_{c \in C_{w_t}} \boldsymbol{v}_c \tag{21}$$

[Mikolov 13] が大きな衝撃をもたらしたのは，単語ベクトルの加減算による類推であろう．特に有名な例は，$\boldsymbol{v}_{\text{king}} - \boldsymbol{v}_{\text{man}} + \boldsymbol{v}_{\text{woman}}$ で求まるベクトルが，$\boldsymbol{v}_{\text{queen}}$ の近くに位置するというものである．これは，「a に対して a^* があるとき，b に対して＿に埋まるものは何か？」という関係類推問題を次式で求めていることに相当する．

$$b^* = \underset{b' \in V}{\text{argmax}} \cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_{a^*} - \boldsymbol{v}_a + \boldsymbol{v}_b) \tag{22}$$

[Levy 14a] は，すべての単語ベクトルが正規化されている場合*10は，式（22）は次式で書き換えられると説明した．

$$b^* = \underset{b' \in V}{\text{argmax}} (\cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_{a^*}) - \cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_a) + \cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_b)) \tag{23}$$

したがって，先ほどの queen を単語ベクトルの加減算で類推することは，「king から近く，woman とも近く，man から遠い場所にある単語は何か？」という問題を解いていることになる．単語ベクトルの加減算による類推に関して，式（23）は一定の説明を与えており，興味深い．

また，式（23）では $\cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_{a^*}) + \cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_b)$ という和を計算しているため，a^* と b の両方ではなく，片方だけに非常に近い単語が b^* として選ばれることがある．この弱点を克服するため，[Levy 14a] は類似度の加減算ではなく乗除算に基づく手法（3CosMul）を提案した．

$$b^* = \underset{b' \in V}{\text{argmax}} \frac{\cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_{a^*}) \cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_b)}{\cos(\boldsymbol{v}_{b'}, \boldsymbol{v}_a) + \varepsilon} \tag{24}$$

ここで，ε は分母が 0 になることを防ぐ定数である．

ところで，[Levy 14b] は式（19）の目的関数を変形することで，次式が導出されることを示した．

$$m_{i,j}^{\text{SGNS}} = \boldsymbol{v}_i \cdot \tilde{\boldsymbol{v}}_j = \text{PMI}(i,j) - \log K \tag{25}$$

すなわち，SGNS の目的関数は単語 i と文脈 j の自己相互情報量 PMI (i, j) を $\log K$ だけシフトした単語文脈行列 $m_{i,j}^{\text{SGNS}}$ を再構成できるように，単語ベクトル $\boldsymbol{v}_i$ と

*9 [Mikolov 13] ではユニグラム確率の 0.75 乗を採用している．

*10 実際，[Mikolov 13] はすべての単語ベクトルを正規化している．

表 1 ice と steam に対する文脈語 j の条件付き確率 $p(j|i)$ と，その比をとった例 [Pennington 14]. 比をとることで，ice に特徴的な文脈（solid）や steam に特徴的な文脈（gas）が際立つ

値	$j=$ solid	$j=$ gas	$j=$ water	$j=$ fashion
$p(j\|\text{ice})$	.000190	.000066	.0030	.000017
$p(j\|\text{steam})$	.000022	.000780	.0022	.000018
$\frac{p(j\|\text{ice})}{p(j\|\text{steam})}$	8.9	0.085	1.36	0.96

文脈ベクトル $\tilde{\boldsymbol{v}}_j$ を学習すると解釈できる．これは，式（2）の単語文脈行列でも，SGNS と同様の性質をもつ単語ベクトルが学習できる可能性を示唆しており，[Levy 14b] はこれを実験的に確認した．なお，[Keerthi 15] も SGNS の目的関数を解析し，SGNS と式（25）の共通点・相違点をより詳細に説明している．

2·6 GloVe

[Pennington 14] は，単語ベクトルの加減算がもつべき効果を出発点として，GloVe というモデルを導出した．ある単語 i の文脈 j の分布を条件付き確率分布 $p(j|i)$ で，別の単語 k の文脈の分布を $p(j|k)$ で表す[*11]．単語 k と対比したとき，単語 i を特徴付ける文脈を $p(j|i)/p(j|k)$ で重み付けるのは自然な仮定である（表 1 の例参照）．また，単語 i と k のベクトルをそれぞれ，$\boldsymbol{v}_i$，$\boldsymbol{v}_k$ で表すことにすると，単語 k との対比において単語 i を特徴付けるベクトルを $\boldsymbol{v}_i-\boldsymbol{v}_k$ で求めるのも自然な仮定である．文脈 j のベクトル $\tilde{\boldsymbol{v}}_j$ との内積を導入し，単語ベクトルと確率の比を関数 F で対応付ける．

$$F((\boldsymbol{v}_i-\boldsymbol{v}_k)\cdot\tilde{\boldsymbol{v}}_j)=\frac{p(j|i)}{p(j|k)} \tag{26}$$

ここで，関数 F として，実数の加法がつくる群から正の実数の積がつくる群への準同形写像 exp を採用すると，

$$\exp((\boldsymbol{v}_i-\boldsymbol{v}_k)\cdot\tilde{\boldsymbol{v}}_j)=\frac{\exp(\boldsymbol{v}_i\cdot\tilde{\boldsymbol{v}}_j)}{\exp(\boldsymbol{v}_k\cdot\tilde{\boldsymbol{v}}_j)}=\frac{p(j|i)}{p(j|k)} \tag{27}$$

ゆえに，

$$\exp(\boldsymbol{v}_i\cdot\tilde{\boldsymbol{v}}_j)=p(j|i)=\frac{\#(i,j)}{\#(i,*)} \tag{28}$$

両辺の対数をとり，$\log\#(i,*)$ に対応するバイアス項を b_i で表し，さらに i と j の対称性を確保するため，j に関するバイアス項 $\tilde{b}_j$ を導入すると，次式が得られる．

$$\boldsymbol{v}_i\cdot\tilde{\boldsymbol{v}}_j=\log\#(i,j)-b_i-\tilde{b}_j \tag{29}$$

すべての $i,j\in V\times C$ に関して，式（29）を満たすように単語ベクトル $\boldsymbol{v}_i$，文脈ベクトル $\tilde{\boldsymbol{v}}_j$，バイアス項 b_i および $\tilde{b}_j$ を決定したい．ところが，学習コーパスから求めた単語 i と文脈 j の共起頻度 $\#(i,j)$ はロングテールな分布となる．共起しない事象やまれに共起する事象を単語ベクトルや文脈ベクトルで再現しようとするのは非合理的である．そこで，[Pennington 14] らは重み付き最小二乗法に基づく目的関数を提案した．

$$\sum_{i=1}^{|V|}\sum_{j=1}^{|C|}f(\#(i,j))\left(\boldsymbol{v}_i\cdot\tilde{\boldsymbol{v}}_j+b_i+\tilde{b}_j-\log\#(i,j)\right)^2 \tag{30}$$

$$f(x)=\begin{cases}(x/\kappa)^\beta & (x<\kappa)\\ 1 & (\text{それ以外の場合})\end{cases} \tag{31}$$

ただし，κ は最小二乗法の重みを 1 から $(x/\kappa)^\beta$ に切り換えるタイミング（頻度）を指定するパラメータ（例えば $\kappa=100$），β は乗数パラメータ（例えば $\beta=0.75$）である．式（31）により，共起頻度の小さい単語・文脈ペアは，最小二乗法の重みが小さくなる．

2·7 単語の分散表現の評価

単語の分散表現の性能は，単語類似度（例えば WordSim 353[*12]），類推（例えば Google データセット[*13]），固有表現抽出，選択選好などのタスクにおける貢献度合いで評価する．標準的な学習コーパスやデータセット，および公開された実装を用いることで，手法間の健全な比較が期待される．

一方，単語の分散表現の性能差は各モデルの理論的な側面よりも，実装上の細かいトリックや実験設定に左右されるとの指摘がある．[Levy 15] は，これまでに紹介した手法・実装が採用しているトリックや実験設定をまとめた（表 2）．文脈の重み付けは，単語から離れている文脈の重みを減らし，単語から近い文脈を重視する．サブサンプリングは，頻出する単語を学習データから確率的に削除し，頻出語が学習に与える影響を削減する．単語分布の補正は，擬似負例をサンプリングする際に用い

表 2 単語ベクトルの学習時に用いられるトリックおよび実験設定. 注[*1] は GloVe に実装されている方式ではなく，word2vec 方式にそろえて実験したことを表す．注[*2] は式（25）の K を設定し，負例サンプリングを模擬することを表す．注[*3] は，式（2）の分母を 0.75 乗することで模擬する

トリック／実験設定	PPMI	SVD	SGNS	GloVe
（前処理）				
文脈幅 ($\delta\in\{2,5,10\}$)	o	o	o	o
文脈の重み付け	o	o	o	o[*1]
サブサンプリング	o	o	o	o
低頻度語の削除	o	o	o	o
（単語と文脈の相関の調整）				
擬似負例数 ($K\in\{1,5,15\}$)	o[*2]	o[*2]	o	
単語分布の補正	o[*3]	o[*3]	o	
（後処理）				
ベクトルの和 ($\boldsymbol{v}_w+\tilde{\boldsymbol{v}}_w$)		o	o	o
特異値乗数 ($\gamma\in\{0,0.5,1\}$)		o		
正規化	o	o	o	o

*11 記法に一貫性をもたせるため，元論文 [Pennington 14] における j と k を入れ換えて説明する．

*12 http://www.cs.cmu.edu/~mfaruqui/suite.html

*13 http://word2vec.googlecode.com/svn/trunk/questions-words.txt

る単語ユニグラム分布の補正（0.75乗）である．ベクトルの和（$\boldsymbol{v}_w+\tilde{\boldsymbol{v}}_w$）は，学習後に単語ベクトルと文脈ベクトルの和をとり，学習結果を安定化させる．

さらに，[Levy 15] は各手法のトリックや設定をそろえて実験を行った．その結果，例えば，単語分布の補正は効果的，特異値の乗数パラメータγは0か0.5がよい，SGNSでは擬似負例数Kを増やしたほうがよい，PPMIでは$K>1$に設定しても効果が薄い，などの推奨設定を紹介している．また，トリックや実験設定をそろえると，手法間に明確な差がないことを報告している．[Schnabel 15] も同様に，分散表現の優劣は評価に用いたタスクや実験設定に左右されることを指摘している．

2·8 その他の研究動向

これまでに紹介した手法は，各単語に単語ベクトルを一つだけ割り当てていた．このため，多義性をもつ単語，例えば*bank*（銀行と土手）や*plant*（植物と工場）のベクトルは，すべての意味を平均化したものになる．この欠点に対処するため，各単語に複数の意味ベクトルを割り当て，学習時に語義目録（word sense inventory）を自動推定しながら意味ベクトルを学習する手法が提案されている [Cheng 15, Li 15a, Neelakantan 14, Tian 14]．例えば [Neelakantan 14] は，SGモデルの学習時に単語の意味を文脈から推定し，パラメータを更新する意味ベクトルを適応的に選択している．

学習コーパスから自動推定した語義目録ではなく，WordNetやFrameNetなどの語彙データベースを活用する研究もある [Faruqui 15a, Jauhar 15, Rothe 15]．[Jauhar 15] は，WordNetの単語はlexemeの和（例えばbloomの単語ベクトルはbloom(organ)とbloom(period)のlexemeベクトルの和），synsetはlexemeの和（例えばflower-bloom-blossom (organ)というsynsetベクトルはflower (organ)，bloom (organ)，blossom (organ)というlexemeベクトルの和）という制約付きでsynset/lexemeのベクトルを学習し，語義曖昧性解消タスクでの最高精度を報告している．これらの研究は，既存の語彙データベースから制約を取り出しているが，語彙データベースの項目の分散表現を獲得していると見ることもできる．

単語文脈行列に種々の行列因子分解（matrix factorization）を適用する研究も発表されている．正準相関分析（CCA）によるアプローチでは，単語の出現と文脈の出現を確率変数とみなし，この二つの確率変数間の相関係数が最も大きくなるような写像空間を求める [Arora 15, Dhillon 15, Rastogi 15, Stratos 15]．[Stratos 15] は，CCAの厳密解がSVDで求まることに基づき，次式の単語文脈行列に対するSVDの理論的な裏付けを説明した．

$$m_{i,j}^{\mathrm{CCA}}=\frac{\#(i,j)^{1/2}}{\sqrt{\#(i,*)^{1/2}\#(*,j)^{1/2}}} \qquad (32)$$

[Rastogi 15] は，CCAを一般化正準相関分析（GCCA）に拡張し，訳語関係，係り受け関係，FrameNetなどの46個の「視点」を活用した多視点学習（multi-view learning）に発展させた．[Li 15b] は，重み付き低ランク半正定行列近似（weighted low-rank positive semidefinite approximation）で単語ベクトルを学習する手法を提案した．

単語ベクトルの学習にスパース符号化（sparse coding）を応用する研究もある [Faruqui 15b, Murphy 12, Yogatama 15]．スパース符号化は，画像や音声などの入力を少数の基底ベクトルの重み付き線形結合に分解するもので，大脳新皮質の一次視覚野（V1）との関連が指摘されている [Olshausen 97]．単語ベクトルの学習では，単語文脈行列$M\in\mathbb{R}^{m\times n}$を疎な行列$A\in\mathbb{R}^{m\times d}$と，密な辞書行列$D\in\mathbb{R}^{d\times n}$の積に分解する．疎な行ベクトル$A_i\in\mathbb{R}^d$を単語$i$の分散表現とすると，$M_i\approx A_iD$である．すなわち，単語文脈行列の行ベクトル$M_i$を，辞書行列$D$が表現する基底ベクトルの重み付き和で近似している．スパース符号化を適用すると，単語ベクトルの各次元を人間が解釈しやすくなる [Murphy 12] だけでなく，単語ベクトルの品質も向上するとの報告もある [Faruqui 15b]．スパース符号化の代表的な定式化を示す．

$$\operatorname*{argmin}_{A,D}\|M-AD\|_2^2+\lambda\|A\|_1+\mu\|D\|_2^2 \qquad (33)$$

すなわち，行列AにL_1正則化（係数λ），行列DにL_2正則化（係数μ）をかけつつ，積ADでMを再構成するという最小二乗法を解いている．[Murphy 12] は，PPMIで定義された単語文脈行列Mに対して，行列Aの要素がすべて非負という制約付きのスパース符号化を行った．[Faruqui 15b] は，非負制約付きのスパース符号化で求めた行列Aに対し，非零の要素を強制的に1とすることで，二値ベクトルで表される単語の分散表現を求めた．[Yogatama 15] は，行列Aの正則化を階層的に構成し，単語ベクトルの次元に階層構造をもたせている．

3. 句や文の分散表現

本章では，単語（または文字）の分散表現から句や文の分散表現を合成し，文書分類，評判分析，機械翻訳などの応用タスクをNNでモデル化する研究を紹介する．なお，前章で導入した記法は無効とする．説明を簡潔にするため，ベクトルは太字にせず，小文字でxなどと記す．添字付きのベクトルx_pは，pに依存したベクトルという意味である（ベクトルxのp番目の要素という意味ではない）．行列は大文字でXなどと記し，添字付きの行列X_pは，pに依存した行列を表す（行列の列ベクトルではない）．単語の分散表現ベクトルの次元数をdとする．NNの構成から明白な場合は，ベクトルの次元数や行列のサイズの説明を省略することがある．

3·1 句の分散表現

まず，複合語などの句の分散表現を学習する方法について考える．最も単純な方法は，複合語を 1 単語とみなし，2 章で説明した手法をそのまま適用することである．実際，[Mikolov 13] は前処理で単語間のコロケーション（結び付き）の強さを測定し，*New York Times* などの複合語を認識してから単語ベクトルを学習している．しかし，この方法はデータ疎問題（data sparseness problem）に陥りやすく，分散表現の品質の低下や未知語問題（学習時に出現しなかった単語の組合せ）を招く．

代わりに，句を構成する単語の分散表現を合成して，句の分散表現を得ることを考える．これは，句や文の意味に関して**意味的構成性**（semantic compositionality）を仮定し，合成手続きを検討していることに相当する．代表的な合成手続きは，ベクトルの和や要素ごとの積である [Mitchell 10]．すなわち，単語 p と q の分散表現をそれぞれ，x_p と x_q とすると，句 pq の分散表現は式（34）または式（35）で計算される（$\odot$ はベクトルの要素ごとの積）．

$$x_{pq}^{(\mathrm{add})} = x_p + x_q \tag{34}$$

$$x_{pq}^{(\mathrm{multi})} = x_p \odot x_q \tag{35}$$

特に，式（34）は句ベクトルを合成するシンプルかつ強力なベースライン手法としてよく用いられる．[Tian 15] は，句ベクトルを単語ベクトルの平均で近似する際の誤差を理論的に解析し，加法構成性のメカニズムを解明した．

3·2 再帰的ニューラルネットワーク

再帰的ニューラルネットワーク（Recursive Neural Network：RecNN）は，木構造の葉から根に向かって分散表現の合成を再帰的に繰り返す NN である*14．文を句構造木や依存構造木で表現し，その木に沿って RecNN を構成すると，統語構造を考慮した句ベクトルが得られる．RecNN では，子ノード p, q の分散表現 x_p, x_q を入力として，その親ノード pq の分散表現 x_{pq} を合成していく．

[Socher 11] は，行列 $W_x \to \mathbb{R}^{d\times 2d}$ に分散表現 x_p, x_q を連結したベクトル（$2d$ 次元）を適用し，分散表現 x_{pq}（d 次元ベクトル）を合成する手法を提案した．

$$x_{pq} = g\left(W_x \begin{bmatrix} x_p \\ x_q \end{bmatrix} + b_x\right) \tag{36}$$

ただし，$b_x \in \mathbb{R}^d$ はバイアス項，g は tanh などの活性化関数である．図 5（a）に，*very good movie* という表現に対して RecNN を構成する例を示した．

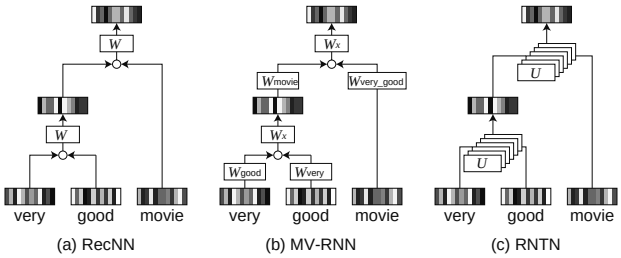

図 5 RecNN，MV-RNN，RNTN で *very good movie* という表現の分散表現を求める例．*very good movie* の句構造は，*very good* という形容詞句と *movie* という名詞で構成されている．したがって，(1) *very* と *good* の分散表現を合成して *very good* の分散表現を得る，(2) さらに *movie* の分散表現と合成する，という手続きで全体の分散表現を求める．○印はベクトルの連結を表す．バイアス項は省略した．MV-RNN では，行列 $X_{\text{very good}}$ も式（39）の合成によって求める．RNTN ではテンソル U による合成（図示）に加えて，RecNN による合成を併用している

RecNN をグラフに展開すると，RecNN を多層 NN とみなすことができる．ゆえに，葉ノードの分散表現，行列，バイアス項は，根·内部ノードに対する教師信号や，オートエンコーダ（autoencoder）による誤差逆伝播法*15 で学習できる．例えば，根ノードの分散表現 x_r から文全体をポジティブ・ネガティブ・ニュートラルの 3 クラスに分類する確率ベクトル y を求めるには，次式の層を追加すればよい．

$$y = \mathrm{softmax}\left(W_{yx}\, x_r + b_y\right) \tag{37}$$

ここで，$W_{yx} \in \mathbb{R}^{3\times d}$，$b_y \in \mathbb{R}^3$ はバイアス項，softmax はソフトマックス関数である．

式（36）は，共通の行列 W で分散表現の合成を行うため，単語・句の意味や統語的な役割に応じた合成を実現できない．また，*extremely amazing* という句では，*extremely* が *amazing* の分散表現を正負両方向に増幅するような合成を期待したいが，式（36）の定式化ではこれが不可能である．そこで，[Socher 12] は各ノードに分散表現 x と合成行列 X（$d\times d$ 行列）をもたせる Matrix-Vector Recursive Neural Network（MV-RNN）を提案した．MV-RNN では，二つの子ノード p, q の分散表現 x_p, x_q，合成行列 X_p, X_q を入力として，その親ノード pq の分散表現 x_{pq} と合成行列 X_{pq} を同時に求める．

$$x_{pq} = g\left(W_x \begin{bmatrix} X_q x_p \\ X_p x_q \end{bmatrix}\right) \tag{38}$$

$$X_{pq} = W_X \begin{bmatrix} X_p \\ X_q \end{bmatrix} \tag{39}$$

ここで，$W_x \in \mathbb{R}^{d\times 2d}$，$W_X \in \mathbb{R}^{d\times 2d}$ である．

MV-RNN は分散表現の合成に関して高い柔軟性をも

*14 オートエンコーダで学習するときは，木構造の根から葉に向かって分散表現の分解を繰り返すが，本稿では割愛する．

*15 正確には back propagation through structure と呼ばれる．

つが，学習コーパス中の全単語に個別の合成行列 X を割り当てたため，学習するパラメータが多くなりすぎるという欠点がある．そこで，[Socher 13] は単一のテンソルを用いて分散表現の合成を行う Recursive Neural Tensor Network（RNTN）を提案した．

$$x_{pq} = g\left(\begin{bmatrix} x_p \\ x_q \end{bmatrix}^{\mathsf{T}} U^{[1:d]} \begin{bmatrix} x_p \\ x_q \end{bmatrix} + W \begin{bmatrix} x_p \\ x_q \end{bmatrix}\right) \quad (40)$$

ここで，$U^{[1:d]} \in \mathbb{R}^{2d \times 2d \times d}$，$W \in \mathbb{R}^{2d \times d}$ である．式（40）のテンソル $U^{[1:d]}$ との積は，テンソルのスライス $U^{[i]} \in \mathbb{R}^{2d \times 2d}$ に対してスカラ値を計算しており，その計算が $i \in \{1, \cdots, d\}$ に関して繰り返され，d 次元のベクトルが計算されると考えるとわかりやすい．

$$i \text{ 次元の値} = \begin{bmatrix} x_p \\ x_q \end{bmatrix}^{\mathsf{T}} U^{[i]} \begin{bmatrix} x_p \\ x_q \end{bmatrix} \quad (41)$$

したがって，式（40）はテンソル $U^{[1:d]}$ に基づくベクトルの合成と，式（36）の RecNN によるベクトルの合成を組み合わせたものと解釈できる．Stanford Sentiment Treebank[*16] による評価実験では，RNTN が RecNN と MV-RNN を上回る性能を示した [Socher 13]．

なお，統語情報を考慮するその他のアプローチとして，MV-RNN の合成行列を単語ではなく品詞で一般化する研究 [Tsubaki 13] や，述語項構造に沿ってベクトルの合成を行う研究 [Hashimoto 14]，品詞や単語クラスタなどの素性値で重み付けを行う研究 [Yu 15]，さまざまな合成関数を実験的に比較した研究 [Muraoka 14] などがある．

3·3 リカレントニューラルネットワーク

リカレントニューラルネットワーク（Recurrent Neural Network：RNN）は，入力系列の分散表現を隠れ層で再帰的に合成しながら，出力系列を予測する NN である．T 個の要素からなる系列に関して，入力のベクトル列 $x_1, \cdots, x_T$ から出力ベクトル列 $y_1, \cdots, y_T$ を予測したい．時刻（位置）t の入力ベクトルを x_t，隠れ状態ベクトルを h_t，出力ベクトルを y_t とすると[*17]，RNN は次式で表される [Sutskever 11]．

$$h_t = g(W_{hx} x_t + W_{hh} h_{t-1} + b_h) \quad (42)$$

$$y_t = W_{yh} h_t + b_y \quad (43)$$

すなわち，時刻 t の隠れ状態ベクトル h_t は，行列 W_{hx} と入力ベクトル x_t の積と，行列 W_{hh} と直前の隠れ状態ベクトル h_{t-1} の積の線形結合に，活性化関数 g（tanh など）を適用したものである．時刻 t の出力 y_t は，隠れ状態ベクトル h_t に適当な変換（式（43）の例では行列 W_{yh}）を施すことによって得られる．なお，b_h と b_y はバイアス項である．時刻 $t = 1$ では，$W_{hh} h_0$ の代わりに，初期バイアス b_{init} を用いる．図6に，*very good movie* という表現に対して RNN を構成する例を示した．RNN の合成を先頭から末尾だけでなく末尾から先頭の双方向にしたり（図7），ある層の隠れ状態を次の層の入力として多層にするなどの拡張もある [Graves 13]．

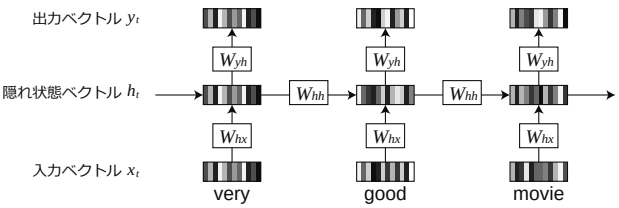

図6 RNN で *very good movie* という表現の分散表現を求める例

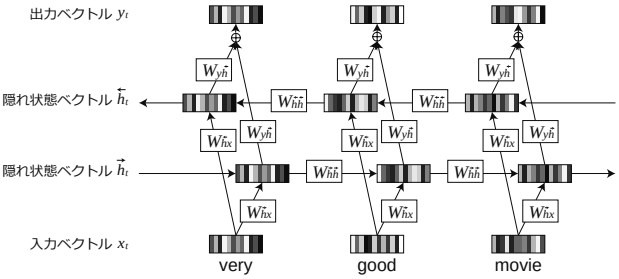

図7 双方向 RNN の構成例

RecNN と同様に，RNN も時刻 t に関してグラフに展開すると，深い NN の一種とみなすことができる．したがって，RNN の学習にも誤差逆伝播法[*18] を適用できる．しかし，確率的勾配降下法と誤差逆伝播法で RNN を学習すると，長期依存（long-range dependency）をうまく扱えない．これは，時間の逆方向に伝播していく勾配（教師信号との誤差）が，指数関数的に消失したり発散するためで，**勾配消失問題**と呼ばれる [Hochreiter 01]．

この問題の解決策の一つが，**長・短期記憶**（Long Short-Term Memory：LSTM）[Hochreiter 97] である[*19]．RNN との決定的な違いは，隠れ状態に加えて記憶セル（memory cell）を導入し，隠れ状態を長期間伝播できるようにしたことである．記憶セルは，入力ゲート（input gate），出力ゲート（output gate），忘却ゲート（forget gate）をもち，それぞれ，入力から記憶セルに書き込む量，記憶セルから出力する量，直前の時刻の記憶セルの内容を保持する量を調整している（図8）．時刻 t の入力 x_t に対して，入力ゲート i_t，忘却ゲート f_t，記憶セル c_t，出力ゲート o_t，隠れ状態 h_t は次式で計算される．

---

*16 http://nlp.stanford.edu/sentiment/index.html

*17 隠れ状態ベクトル h_t や出力ベクトル y_t の次元は任意（ハイパーパラメータ）であり，次元が合うように行列 W_{hx}, W_{hh}, W_{yh} のサイズを決定すればよい．

*18 正確には back propagation through time と呼ばれる．

*19 本稿では [Graves 13] に基づいて説明する．

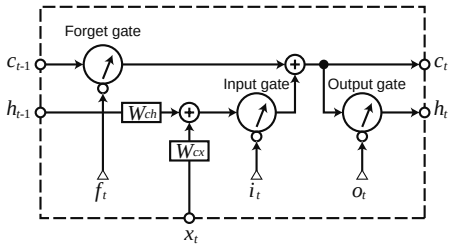

図 8　Long Short-Term Memory（LSTM）．
f_t, i_t, o_t を計算する部分は省略した（x_t, h_{t-1}, c_{t-1} に基づき，同様の式で計算する）

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + W_{ic}c_{t-1} + b_i) \quad (44)$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + W_{fc}c_{t-1} + b_f) \quad (45)$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + W_{oc}c_{t-1} + b_o) \quad (46)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \quad (47)$$

$$h_t = o_t \odot g(c_t) \quad (48)$$

LSTM は，単語分割 [Chen 15]，係り受け解析 [Dyer 15]，機械翻訳 [Sutskever 14]，意見分析 [Wang 15]，情報抽出 [Xu 15b] など，言語処理への応用が急増している．また，RecNN を LSTM で拡張する研究もある [Tai 15]．

LSTM に似た機構をもち，より簡潔なモデルとして，**ゲート付きリカレントユニット**（Gated Recurrent Unit：GRU）[Cho 14] も有力である．GRU は時刻 t の入力 x_t と直前の隠れ状態 h_{t-1} をもとに，リセットゲート r_t，更新ゲート z_t，隠れ状態 h_t を計算する．

$$r_t = \sigma(W_{rx}x_t + W_{rh}h_{t-1} + b_r) \quad (49)$$

$$z_t = \sigma(W_{zx}x_t + W_{zh}h_{t-1} + b_z) \quad (50)$$

$$\tilde{h}_t = g(W_{hx}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \quad (51)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (52)$$

リセットゲート r_t の値が小さくなると，直前の隠れ状態 h_{t-1} は無視され，隠れ状態は現在の入力 x_t でリセットされる．更新ゲート z_t は，直前の隠れ状態 h_{t-1} を現状態 h_t に引き継ぐ割合を調整する（図 9）．

式（47）と式（52）で表現されるように，LSTM と GRU は時刻 t の隠れ状態ベクトルを計算するときに，時刻 $t-1$ の隠れ状態ベクトルと時刻 t の入力から計算されるベクトルの重み付き和を計算するという機構を有している．この機構のおかげで，重要な特徴を RNN 中で長期間保存したり，離れた時刻の隠れ状態間の短経路が形成され，勾配消失問題が軽減されると考えられている．一方で，LSTM では記憶セルの内容が出力ゲートを介して出力されるが，GRU では記憶内容がそのまま出力される．また，LSTM では入力ゲートと出力ゲートを独立に制御できるが，GRU では更新ゲートが入力ゲートと出力ゲートを同時に制御する，という差異もある．

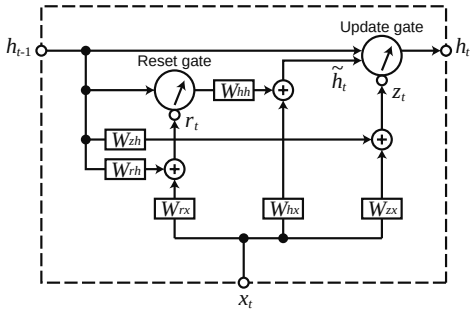

図 9　Gated Recurrent Unit（GRU）．
更新ゲート（update gate）は，h_{t-1} と $\tilde{h}_t$ の配分を z_t で調整する

LSTM や GRU の有用性は多くの研究で実証されているが，それらを構成する各要素の意義は定性的な議論に留まっている．特に，たくさんのゲートで構成される LSTM はアドホックだと批判されることもある．実際に，LSTM よりも GRU のほうが高い性能を示すという報告もある [Chung 14]．このため，LSTM と GRU を比較したり [Karpathy 15]，LSTM で必須の要素や [Greff 15]，LSTM や GRU の亜種を探索する研究も進められている [Jozefowicz 15]．

3·4　畳込みニューラルネットワーク

物体認識で大成功を収めた**畳込みニューラルネットワーク**（Convolutional Neural Network：CNN）[LeCun 98] も，言語処理での応用が急増している．二次元に画素が配置される画像とは異なり，言語では一次元の畳込みを考える．

T 個の要素からなる単語列のベクトルを $x_1, \cdots, x_T$，時刻 t の単語ベクトルを $x_t \in \mathbb{R}^d$ と書く．時刻 t から $t+\delta-1$ の単語のベクトルを連結したものを時刻 t の領域ベクトルと呼び，$x_{t:t+\delta} \in \mathbb{R}^{d\delta}$ で表す．

$$x_{t:t+\delta} = x_t \oplus x_{t+1} \oplus \ldots \oplus x_{t+\delta-1} \quad (53)$$

式（53）を時刻 $t \in \{1, \cdots, T-\delta+1\}$ に関して繰り返し適用することで，時刻をずらしながら δ-gram に対応するベクトルを得る[*20]．

時刻 t の領域ベクトル $x_{t:t+\delta}$ に次式を適用し，時刻 t における特徴量 $p_t \in \mathbb{R}$ を得る．

$$p_t = g(w \cdot x_{t:t+\delta} + b) \quad (54)$$

*20　ここで説明した領域ベクトルのつくり方は，狭い畳込み（narrow convolution）と呼ばれる [Kalchbrenner 14]．これに対し，系列の先頭と末尾に零ベクトルを $\delta-1$ 個ずつ挿入し（zero padding），$T+\delta-1$ 個の領域ベクトルを取る方式は，広い畳込み（wide convolution）と呼ばれる．

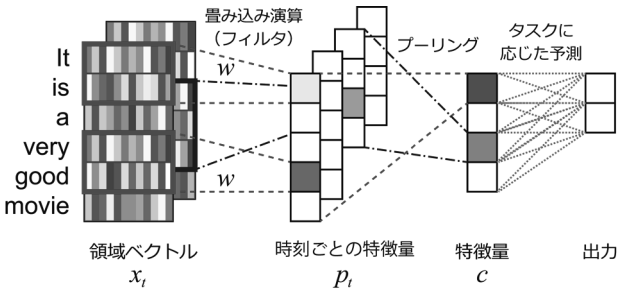

図 10 CNN の構成例

ただし，パラメータ $w \in \mathbb{R}^{d\delta}$, $b \in \mathbb{R}$ は時刻によらず，共通のものを用いる．式（54）の処理はフィルタ（filter）と呼ばれ，この処理を $t \in \{1, \cdots, T-\delta+1\}$ に関して繰り返すと，**畳込み演算**（convolution operation）となる．

式（54）で作成した特徴量の数は入力テキストの長さ T に応じて変わる．これを固定長の特徴量 $c \in \mathbb{R}$ に変換し，タスクを解く NN に接続する．この変換の代表例が**最大値プーリング**（max pooling）である．

$$c = \max_{1<t\leq T-\delta+1} p_t \tag{55}$$

最大値プーリングは，時刻によらずに入力の中で最も「重要」な特徴を見いだすことに相当する．他にも，各時刻の特徴量 p_t の平均値を系列全体の特徴量とする平均値プーリング（average pooling），最大値だけではなく上位 k 個の特徴量を取り出す k 最大値プーリング（k-max pooling），k 最大値プーリングにおける k の値を入力長に応じて適応的に調整する動的 k 最大値プーリング（dynamic k-max pooling）などが提案されている [Kalchbrenner 14].

式（55）で得られた特徴量 c はスカラ値であるが，領域ベクトルの取り方やフィルタのパラメータ，プーリングの方法を変えることで，複数の特徴量を取り出すこともできる（図 10）．例えば，複数の領域幅 δ，単語の分散表現，文字の分散表現を同時に採用することで，異なる種類の特徴を表現する領域ベクトルを取り出せる [Johnson 15, Kim 14, Santos 14]．式（54）の w を $h \times d\delta$ 行列，b を h 次元ベクトルに替えれば，異なる特徴に着目する h 個のフィルタを学習できると期待される [Xu 15a]．最大値プーリングと平均値プーリングを併用したり [Zhang 15]，入力テキストを複数のセグメントに区切って複数のプーリングを取る方法 [Zeng 15] も提案されている．さらに，CNN を構文木上に拡張する研究もある [Ma 15].

3·5 その他の研究動向

単語の分散表現から入力文の分散表現を合成（encode）し，その分散表現を翻訳先の言語で分解（decode）することで，機械翻訳を NN のみで実現するエンコーダ・デコーダ（encoder-decoder）アーキテクチャが提案された [Cho 14, Sutskever 14]．このアーキテクチャの拡張として，ある文の分散表現を GRU などで合成し，その分散表現から周辺の文を予測できるようなデコーダを学習するという Skip-Thoughts モデルが提案された [Kiros 15]．エンコーダとデコーダのパラメータは大量のラベルなしテキストだけで学習できるが，文の類似性判定などのタスクで高い性能が報告されている．

エンコーダ・デコーダアーキテクチャでは，入力文の情報を固定次元の分散表現に押し込めるため，入力文が長いと情報が失われたり，単語を合成・分解する順序の影響を受けやすくなる．長期依存を学習できるとされている LSTM や GRU をエンコーダやデコーダに採用しても，これらの弱点の解消は難しい．そこで，注意機構（attention mechanism）を導入し，デコーダがエンコードすべき箇所を制御する手法が提案された [Bahdanau 15]．注意機構により，エンコーダは入力文のすべての情報を分散表現に詰め込まずに済む．なお，エンコーダ・デコーダアーキテクチャや注意モデルの詳細は，本特集内の渡辺氏の解説記事 [渡辺 16] を参照されたい．

段落や文書という単位の分散表現を学習する手法としては，パラグラフベクトルが有名である [Le 14]．パラグラフベクトルは，SG モデルや CBOW モデルを拡張し，段落内・文書内で共通に更新するベクトルを導入することで，段落・文書の分散表現を学習する．非常にシンプルな手法であるが，感情分析などの分類タスクで安定した性能が報告されている．

Freebase[*21] や DBPedia[*22] などの知識ベースの分散表現を学習する研究も進められている．知識ベースに収録されている実体や関係の分散表現を学習することで，テキストから関係事例（relation instance）を抽出したり，知識ベースに収録されていない関係事例を推論することが可能となる．[Takase 16] は，名詞句対の関係を表す言語パターン（動詞句など）の分散表現を RecNN で合成し，そのパラメータを SGNS で学習することで，加法構成性よりも質の良い分散表現が獲得できることを示した．[Toutanova 15] は，単語の依存関係パスで表現された関係パターン

$$\left(\text{例}: x \xleftarrow{\text{nsubj}} \text{president} \xrightarrow{\text{prep}} \text{of} \xrightarrow{\text{obj}} y \right)$$

の分散表現の合成を CNN でモデル化しつつ，知識ベースと関係パターンの分散表現を同一のベクトル空間で学習する手法を提案した．[Neelakantan 15] は，`IsBasedIn`(a, b)，`StateLocatedIn`(b, c)，`CountryLocatedIn`(c, d) のような関係の連鎖から `CountryOfHeadquaters`(a, d) を推論するタスクを，関係の分散表現を RecNN で繰り返し合成するというアイディアで実現した．[Guu 15] も，関係の連鎖を関係行列の合成でモデル化する手法を提案し，知識ベースへの問合せに対する答えを求めるタスク

*21 https://www.freebase.com/
*22 http://wiki.dbpedia.org/

や，知識ベースのエントリを補完するタスクにおいて，劇的な性能向上を報告している．

[Rocktäschel 15] は，一階述語論理で記述された制約を考慮した行列因子分解を提案した．この研究では，エンティティペアと関係の分散表現を学習する際に，$\forall x, y$: daughter-of$(x, y) \Rightarrow$person/parents(x, y) のような外部知識を埋め込むことができる．これにより，述語論理による制約を考慮しなくても，分散表現に対する内積演算のみで未知の事例を推論できる点が興味深い．

[Yang 15] は，エンティティや関係の分散表現を NN で学習するさまざまな手法を比較・分析している．また，知識ベースにおける分散表現の応用については，サーベイ論文 [Nickel 16] が詳しい．

4. お わ り に

本稿では，言語処理における分散表現学習の最新動向を紹介した．画像処理や音声処理では入力信号の特徴記述が自明ではなく，深層学習が革新的な成果を収めたが，言語処理では記号による特徴記述（例えば bag-of-words 表現）を超える研究の登場が遅れた．そんな中，単語の分散表現の研究が開花したことで，単語，句，文，文書，知識ベースなどの分散表現を NN で自由に設計し，複雑な言語現象や言語処理タスクに迫ろうとする研究が次々と発表されている．

謝 辞

本稿の執筆にあたり，英リバプール大学の Danushka Bollegala 氏，東北大学の乾健太郎氏，田 然氏，小林颯介氏，横井 祥氏より有益なご助言をいただきました．この場を借りて感謝申し上げます．

◇ 参 考 文 献 ◇

[Arora 15] Arora, S., Li, Y., Liang, Y., Ma, T. and Risteski, A.: RAND-WALK: A latent variable model approach toword embed dings, *CoRR,* Vol. abs/1502.03520（2015）

[Bahdanau 15] Bahdanau, D., Cho, K. and Bengio,Y.: Neural machine translation by jointly learning to align and translate, *ICLR*（2015）

[Bengio 03] Bengio, Y., Ducharme, R.,Vincent, P. and Janvin, C.: A neural probabilistic language model, *J. Machine Learning Research,* Vol. 3, pp. 1137-1155（2003）

[Bengio 09] Bengio, Y., Louradour, J., Collobert, R. and Weston, J.: Curriculum learning, *Proc. of ICML,* pp. 41-48（2009）

[Bullinaria 07] Bullinaria, J. A. and Levy, J. P.: Extracting semantic representations from word co-occurrence statistics: A computational study, *Behavior Research Methods,* Vol. 39, No. 3, pp. 510-526（2007）

[Chen 15] Chen, X., Qiu, X., Zhu, C., Liu,P. and Huang, X.: Long short-term memory neural networks for Chinese word segmentation, *Proc. of EMNLP,* pp. 1197-1206（2015）

[Cheng 15] Cheng, J. and Kartsaklis, D.: Syntax-aware multi-sense word embeddings for deep compositional models of meaning, *Proc. of EMNLP,* pp. 1531-1542（2015）

[Cho 14] Cho, K., Merrienboer, van B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation, *Proc. of EMNLP,* pp. 1724-1734（2014）

[Chung 14] Chung, J.,Gülçehre, Ç., Cho, K. and Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling, *CoRR,* Vol. abs/1412.3555（2014）

[Collobert 08] Collobert, R. and Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning, *Proc. of ICML,* pp. 160-167（2008）

[Collobert 11] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P.: Natural language processing（almost）from scratch, *J. Machine Learning Research,*Vol. 12, pp. 2493-2537（2011）

[Deerwester 90] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R.: Indexing by latent semantic analysis, *J. American Society for Information Science,*Vol. 41, No. 6, pp. 391-407（1990）

[Dhillon 15] Dhillon, P. S., Foster, D. P. and Ungar, L. H.: Eigenwords: Spectral word embeddings, *J. Machine Learning Research,* Vol. 16, pp. 2999-3034（2015）

[Dyer 15] Dyer, C., Ballesteros, M., Ling, W., Matthews, A. and Smith, N. A.: Transition-based dependency parsing with stack long short-term memory, *Proc. of ACL-IJCNLP,* pp. 334-343（2015）

[Faruqui 15a] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E. and Smith, N. A.: Retrofitting word vectors to semantic lexicons, *Proc. of NAACL-HLT,* pp. 1606-1615（2015）

[Faruqui 15b] Faruqui, M., Tsvetkov, Y.,Yogatama, D., Dyer, C. and Smith,N. A.: Sparse overcomplete word vector representations, *Proc. of ACL-IJCNLP,* pp. 1491-1500（2015）

[Firth 57] Firth, J. R.: *Papers in Linguistics 1934-1951*, Oxford University Press, London（1957）

[Goldberg 15] Goldberg, Y.: A Primer on neural network models for natural language processing, *CoRR,* Vol. abs/1510.00726（2015）

[Graves 13] Graves, A.: Generating sequences with recurrent neural networks, *CoRR,* Vol. abs/1308.0850（2013）

[Greff 15] Greff, K., Srivastava, R. K.,Koutník, J., Steunebrink, B. R. and Schmidhuber, J.: LSTM: A search space odyssey, *CoRR,* Vol. abs/1503.04069（2015）

[Guu 15] Guu, K., Miller, J. and Liang, P.: Traversing knowledge graphs in vector space, *Proc. of EMNLP,* pp. 318-327（2015）

[Halko 11] Halko, N., Martinsson, P.-G. and Tropp, J. A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Review,* Vol. 53, No. 2, pp. 217-288（2011）

[Harris 54] Harris, Z.: Distributional structure, *Word,* Vol. 10, No. 23, pp. 146-162（1954）

[Hashimoto 14] Hashimoto, K., Stenetorp, P., Miwa, M. and Tsuruoka, Y.: Jointly learning word representations and composition functions using predicate-argument structures, *Proc. of EMNLP,* pp. 1544-1555（2014）

[Hinton 86] Hinton, G., McClelland, J. and Rumelhart, D.: Distributed representations, Rumelhart, D. E., McClelland, J. L. and Group, P. R., eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* Vol. I, chapter 3, pp. 77-109, Cambridge, MA, MIT Press（1986）

[Hochreiter 97] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural Computation,* Vol. 9, No. 8, pp. 1735-1780（1997）

[Hochreiter 01] Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J.: Gradient flow in recurrent nets: The difficulty of learning long-term dependencies, Kolen, J. F. and Kremer, S. C., eds., *Field Guide to Dynamical Recurrent Networks,* chapter 14, pp. 237-243, Wiley-IEEE Press（2001）

[Jauhar 15] Jauhar, S. K., Dyer, C. and Hovy, E.: Ontologically grounded multi-sense representation learning for semantic vector space models, *Proc. of NAACL-HLT,* pp. 683-693（2015）

[Johnson 15] Johnson, R. and Zhang, T.: Effective use of word

order for text categorization with convolutional neural networks, *Proc. of NAACL-HLT,* pp. 103-112（2015）
[Jozefowicz 15] Jozefowicz, R., Zaremba,W. and Sutskever, I.: An empirical exploration of recurrent network architectures, *Proc. of ICML,* pp. 2342-2350（2015）
[Kalchbrenner 14] Kalchbrenner, N., Grefenstette, E. and Blunsom, P.: A convolutional neural network for modelling sentences, *Proc. of ACL,* pp. 655-665（2014）
[Karpathy 15] Karpathy, A., Johnson, J. and Li, F.:Visualizing and understanding recurrent networks, *CoRR,* Vol. abs/1506.02078（2015）
[Keerthi 15] Keerthi, S. S., Schnabel, T. and Khanna, R.: Towards a better understanding of predict and count models, *CoRR,* Vol. abs/1511.02024（2015）
[Kim 14] Kim, Y.: Convolutional neural networks for sentence classification, *Proc. of EMNLP,* pp. 1746-1751（2014）
[Kiros 15] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A. and Fidler, S.: Skip-thought vectors, *Proc. of NIPS,* pp. 3276-3284（2015）
[Le 14] Le, Q. and Mikolov, T.: Distributed representations of sentences and documents, *Proc. of ICML,* pp. 1188-1196（2014）
[LeCun 98] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, *Proc. of IEEE,* Vol. 86, No. 11, pp. 2278-2324（1998）
[Levy 14a] Levy, O. and Goldberg, Y.: Linguistic regularities in sparse and explicit word representations, *Proc. of CoNLL,* pp. 171-180（2014）
[Levy 14b] Levy, O. and Goldberg, Y.: Neural word embedding as implicit matrix factorization, *Proc. of NIPS,* pp. 2177-2185（2014）
[Levy 15] Levy, O., Goldberg, Y. and Dagan, I.: Improving distributional similarity with lessons learned from word embeddings, *Trans. of the Association for Computational Linguistics,* Vol. 3, pp. 211-225（2015）
[Li 15a] Li, J. and Jurafsky,D.: Do multi-sense embeddings improve natural language understanding?, *Proc. of EMNLP,* pp. 1722- 1732（2015）
[Li 15b] Li, S., Zhu, J. and Miao, C.: A generative word embedding model and its low rank positive semidefinite solution, *Proc. of EMNLP,* pp. 1599-1609（2015）
[Ma 15] Ma, M., Huang, L., Zhou, B. and Xiang, B.: Dependency-based convolutional neural networks for sentence embedding, *Proc. of ACL-IJCNLP*（Volume 2: Short Papers）, pp. 174-179（2015）
[Mikolov 13] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed representations of words and phrases and their compositionality, *Proc. of NIPS,* pp. 3111-3119（2013）
[Mitchell 10] Mitchell, J. and Lapata, M.: Composition in distributional models of semantics, *Cognitive Science,* Vol. 34, No. 8, pp. 1388-1429（2010）
[Mnih 07] Mnih, A. and Hinton, G.: Three new graphical models for statistical language modelling, *Proc. of ICML,* pp. 641-648（2007）
[Mnih 09] Mnih, A. and Hinton, G. E.: A scalable hierarchical distributed language model, *Proc. of NIPS,* pp. 1081-1088（2009）
[Morin 05] Morin, F. and Bengio, Y.: Hierarchical probabilistic neural network language model, *Proc. of AISTATS,* pp. 246-252（2005）
[Muraoka 14] Muraoka, M., Shimaoka, S., Yamamoto, K., Watanabe, Y., Okazaki, N. and Inui, K.: Finding the best model among representative compositional models, *Proc. of PACLIC,* pp. 65- 74（2014）
[Murphy 12] Murphy, B., Talukdar, P. and Mitchell, T.: Learning effective and interpretable semantic models using non-negative sparse embedding, *Proc. of COLING,* pp. 1933-1950（2012）
[Neelakantan 14] Neelakantan, A., Shankar, J., Passos, A. and McCallum, A.: Efficient non-parametric estimation of multiple embeddings per word in vector space, *Proc. of EMNLP,* pp. 1059-1069（2014）
[Neelakantan 15] Neelakantan, A., Roth, B. and McCallum, A.: Compositional vector space models for knowledge base completion, *Proc. of ACL-IJCNLP,* pp. 156-166（2015）
[Nickel 16] Nickel, M., Murphy, K., Tresp, V. and Gabrilovich, E.: A review of relational machine learning for knowledge graphs, *Proc. IEEE,* Vol. 104, No. 1, pp. 11-33（2016）
[Olshausen 97] Olshausen, B. A. and Field, D. J.: Sparse coding with an over complete basis set: A strategy employed by V1?, *Vision Research,* Vol. 37, No. 23, pp. 3311-3325（1997）
[Pennington 14] Pennington, J., Socher, R. and Manning, C.: Glove: Global vectors for word representation, *Proc. of EMNLP,* pp. 1532-1543（2014）
[Rastogi 15] Rastogi, P., Van Durme, B. and Arora, R.: Multiview LSA: Representation learning via generalized CCA, *Proc. of NAACL-HLT,* pp. 556-566（2015）
[Rocktäschel 15] Rocktäschel, T., Singh, S. and Riedel, S.: Injecting logical background knowledge into embeddings for relation extraction, *Proc. of NAACL-HLT,* pp. 1119-1129（2015）
[Rothe 15] Rothe, S. and Schütze, H.: AutoExtend: Extending word embeddings to embeddings for synsets and lexemes, *Proc. of ACL-IJCNLP,* pp. 1793-1803（2015）
[Salton 75] Salton, G., Wong, A. and Yang, C. S.: A vector space model for automatic indexing, *Commun. of the ACM,* Vol. 18, No. 11, pp. 613-620（1975）
[Santos 14] Santos, dos C. and Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts, *Proc. of COLING,* pp. 69-78（2014）
[Schnabel 15] Schnabel, T., Labutov, I., Mimno, D. and Joachims, T.: Evaluation methods for unsupervised word embeddings, *Proc. of EMNLP,* pp. 298-307（2015）
[Socher11] Socher, R., Pennington, J., Huang, E. H., Ng, A. Y. and Manning, C. D.: Semi-supervised recursive autoencoders for predicting sentiment distributions, *Proc. of EMNLP,* pp. 151-161（2011）
[Socher 12] Socher, R., Huval, B., Manning, C. D. and Ng, A. Y.: Semantic compositionality through recursive matrix-vector spaces, *Proc. of EMNLP-CoNLL,* pp. 1201-1211（2012）
[Socher 13] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. and Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank, *Proc. of EMNLP,* pp. 1631-1642（2013）
[Stratos 15] Stratos, K., Collins, M. and Hsu, D.: Model-based word embeddings from decompositions of count matrices, *Proc. of ACL-IJCNLP,* pp. 1282-1291（2015）
[Sutskever 11] Sutskever, I., Martens, J. and Hinton, G.: Generating text with recurrent neural networks, *Proc. of ICML,* pp. 1017- 1024（2011）
[Sutskever 14] Sutskever, I., Vinyals, O. and Le, Q. V.: Sequence to sequence learning with neural networks, *Proc. of NIPS,* pp. 3104-3112（2014）
[Tai 15] Tai, K. S., Socher, R. and Manning, C. D.: Improved semantic representations from tree-structured long short-term memory networks, *Proc. of ACL-IJCNLP,* pp. 1556-1566（2015）
[Takase 16] Takase, S., Okazaki, N. and Inui, K.: Modeling semantic compositionality of relational patterns, *Engineering Applications of Artificial Intelligence*, Vol. 50, pp. 256-264（2016）
[Tian 14] Tian, F., Dai, H., Bian, J., Gao, B., Zhang, R., Chen, E. and Liu, T.-Y.: A probabilistic model for learning multi-prototype word embeddings, *Proc. of COLING,* pp. 151-160（2014）
[Tian 15] Tian, R., Okazaki, N. and Inui, K.: The mechanism of additive composition, *CoRR,* Vol. abs/1511.08407（2015）
[Toutanova 15] Toutanova,K., Chen,D.,Pantel,P., Poon,H., Choudhury, P. and Gamon, M.: Representing text for joint embedding of text and knowledge bases, *Proc. of EMNLP,* pp. 1499-1509（2015）
[Tsubaki 13] Tsubaki, M., Duh, K., Shimbo, M. and Matsumoto, Y.: Modeling and learning semantic co-compositionality through prototype projections and neural networks, *Proc. of*

EMNLP, pp. 130-140（2013）
[Tsuboi 14] Tsuboi, Y.: Neural networks leverage corpus-wide information for part-of-speech tagging, *Proc. of EMNLP,* pp. 938- 950（2014）
[坪井 15] 坪井祐太：自然言語処理におけるディープラーニングの発展，オペレーションズ・リサーチ：経営の科学，Vol. 60, No. 4, pp. 205-211（2015）
[Turian 10] Turian, J., Ratinov, L.-A. and Bengio, Y.: Word representations: a simple and general method for semi-supervised learning, *Proc. of ACL,* pp. 384-394（2010）
[Turney 10] Turney, P. D. and Pantel, P.: From frequency to meaning: vector space models of semantics, *J. Artificial Intelligence Research,* Vol. 37, pp. 141-188（2010）
[Wang 15] Wang, X., Liu, Y., SUN, C., Wang, B. and Wang, X.: Predicting polarities of tweets by composing word embeddings with long short-term memory, *Proc. of ACL-IJCNLP,* pp. 1343-1353（2015）
[渡辺 16] 渡辺太郎：ニューラルネットワークによる構造学習の発展，人工知能，Vol. 31, No. 2, pp. 202-209（2016）
[Xu 15a] Xu, K., Feng,Y., Huang, S. and Zhao, D.: Semantic relation classification via convolutional neural networks with simple negative sampling, *Proc. of EMNLP,* pp. 536-540（2015）
[Xu 15b] Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H. and Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths, *Proc. of EMNLP,* pp. 1785-1794（2015）
[Yang 15] Yang, B.,Yih, tauW., He, X., Gao, J. and Deng, L.: Embedding entities and relations for learning and inference in knowledge bases, *Proc. of ICLR*（2015）
[Yogatama 15] Yogatama, D., Faruqui, M., Dyer, C. and Smith, N. A.: Learning word representations with hierarchical sparse coding, *Proc. of ICML*（2015）
[Yu 15] Yu, M. and Dredze, M.: Learning composition models for phrase embeddings, *Trans. of the Association for Computational Linguistics,* Vol. 3, pp. 227-242（2015）
[Zeng 15] Zeng, D., Liu, K., Chen,Y. and Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks, *Proc. of EMNLP,* pp. 1753-1762（2015）
[Zhang 15] Zhang, B., Su, J., Xiong, D., Lu, Y., Duan, H. and Yao, J.: Shallow convolutional neural network for implicit discourse relation recognition, *Proc. of EMNLP,* pp. 2230-2235（2015）

2016 年 1 月 18 日　受理

著　者　紹　介

岡崎　直観（正会員）

2007 年東京大学大学院情報理工学研究科博士課程修了．2005 年英国テキストマイニングセンターリサーチフェロー，2007 年東京大学大学院情報理工学系研究科特別研究員を経て，2011 年より東北大学大学院情報科学研究科准教授．専門は自然言語処理，テキストマイニング，機械学習．