Vol.2015-NL-222 No.5
2015/7/15

ニューラルネットワークによる意味構成とそのカーネル埋め込みを用いた多層非線形類似度学習

椿真史[1,a)]　Duh Kevin[1,b)]　新保仁[1,c)]　松本裕治[1,d)]

概要：自然言語処理の様々なアプリケーションにおいて，単語や句，文や文書などの意味的な類似度を適切に計算することは重要である．このような類似度は単語ベクトル空間を用いることで，個々の単語間であればある程度適切に計算することができる．しかし，個々の単語から文全体の意味表現をベクトル空間上でどのように構成し，その類似度をどのように計算するかについては自明ではない．そこで我々は本稿で，ニューラルネットワークによる意味構成とそのカーネル埋め込みを用いた多層非線形類似度学習を提案する．提案法は，カーネル関数によって写像された高次元空間における表現学習であり，文の意味構成にとってより適切な単語表現を新たに獲得することを可能にする．我々はこの提案法を，自然言語処理のコンペティションである SemEval 2014 の文の意味的類似度データセット (Task 1: Sentences Involving Compositional Knowledge) を用いて評価した．その結果，ベースラインである線形手法，Recursive Neural Network (RNN)，そしてコンペティションでの上位チームをすべて上回り，さらに構造化された Long Short-Term Memory (LSTM) と同程度の性能を達成した．

1. はじめに

情報検索やテキストマイニングなどの自然言語処理アプリケーションでは，単語や句，文や文書における類似度が重要となる．そのような類似度を計算する単純な手法として，表層的な素性に基づくもの (例えば 2 つの文書に同じ単語が出現する回数など) がある．しかしこのようなアプローチでは，言語が持つ「意味」を捉えることはできない．

古くから，単語ベクトル空間モデルに基づく意味研究が行われており，近年でも盛んに研究されている [1], [2], [3]．我々はこの単語ベクトル表現を用いることで，個々の単語間の意味的な類似度であればある程度適切に計算することができる．しかし，複数の単語から文全体の意味表現をベクトル空間上でどのように構成し，その類似度をどのように計算するかについては自明ではない．このような問題を解決するため，単語ベクトル空間における意味構成の研究が，特に近年盛んに行われるようになった [4], [5], [6], [7]．この研究の最終的な目標は，句や文の意味を適切に表現するような単語ベクトルに対する関数，つまり意味の構成関数を，適切にモデル化し学習することである．

しかし，単語ベクトル空間モデルにおける意味構成の研究には，未だ多くの問題が存在する．特に本稿で我々は，以下の 2 つの問題に焦点を当てる．

(1) 文の意味をその構造も含めて，ベクトル空間上でどのように表現するのか？
(2) 単語よりも遥かに多くのバリエーションを持つ文の意味を，個々の単語ベクトル表現からどのように構成するのか？

これらの問題を解決するために我々は，文の構造と意味を単語ベクトル空間よりも高い表現力を持つ高次元空間において学習することに焦点を当てる．例えば，*"Newton was inspired to formulate gravitation by watching the fall of an apple from a tree."* という文が表現され得る空間は，*"apple"* や *"gravitation"*，*"formulate"* や *"by"* などの個々の単語が表現されている空間とは異なり，文が持つより複雑な構造と豊かな意味を表現できなければならない．つまり我々は，単語から文の意味構成に伴い，より表現力の高い高次元空間をも同時に構成し学習する必要がある．

本稿で我々は，ニューラルネットワークによる意味構成とそのカーネル埋め込みを用いた多層非線形類似度学習を提案する．我々は，計量学習 [8] の研究の中でも特に，カーネル関数を用いたその非線形拡張 [9] に焦点を当てる．この提案法の概略を図 1 に示す．この図にあるように提

[1] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-0192, Japan
[a)] masashi-t@is.naist.jp
[b)] kevinduh@is.naist.jp
[c)] shimbo@is.naist.jp
[d)] matsu@is.naist.jp

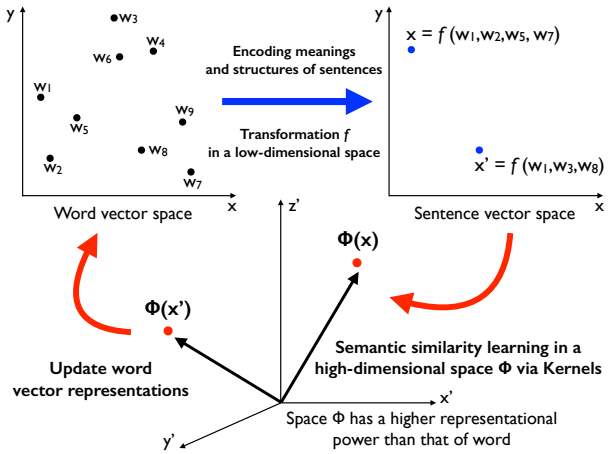

図 1 提案法の概略図．我々は，カーネル関数によって写像される高次元空間が，構成に伴なって生じる複雑な構造と豊かな意味を，よりシンプルに表現し学習できることに着目する．提案法は，文の高次元意味ベクトル表現を陽に計算することなく，カーネルによって写像された高次元空間においてその類似度のみを計算し学習することで，文の意味構成により適した単語表現を新たに獲得する．

案法は，文の高次元意味ベクトル表現を陽に計算することなく，カーネルによって写像された高次元空間においてその類似度のみを計算し学習することで，文の意味構成により適した単語表現を新たに獲得する．また我々のアプローチは，Recursive Neural Network (RNN) [6], [10] や Long Short-Term Memory (LSTM) [11] などの自然言語処理における Deep Learning の手法とは一線を画する．これらの手法は文の構造と意味を，低次元空間において詳細かつ複雑にモデル化するが，提案法は低次元空間と高次元空間の双方を用いることでそれらをよりシンプルに表現し学習することができる (図 2 と 5.4 節参照)．

本稿の貢献は以下の 3 点である．

(1) 低次元空間における構造表現と，高次元空間における意味学習の双方を用いることで，複雑な構造を持つ文の意味構成のための新たな単語表現学習法を提案した．

(2) 提案法は，自然言語処理のコンペティションである SemEval 2014 の文の意味的類似度データセット (Task 1: Sentences Involving Compositional Knowledge) において，ベースラインである線形手法，Recursive Neural Network (RNN)，そしてコンペティションでの上位チームをすべて上回り，さらに構造化された Long Short-Term Memory (LSTM) と同程度の性能を達成した．

(3) 我々は，ベクトル空間での意味構成モデルにおけるニューラルネットワークによる非線形関数の多層適用と，非線形カーネルによる高次元空間への写像という 2 つの手法の非線形性に関して，新たな視点を提供した．

2. 背景

2.1 単語ベクトル空間と文の意味構成モデル

単語をベクトルによって表現する手法はこれまで数多く提案されており，古くは単語の共起頻度行列の特異値分解に基づく潜在意味解析から，近年ではニューラルネットワークを用いたニューラル言語モデルに至るまで，そのアプローチは様々である [1], [2], [3], [12], [13], [14], [15], [16]．これらはアプローチの違いこそあれど，得られた単語ベクトル間の距離や内積を計算することで，単語間の意味的な類似度をある程度適切に計算することができるという点においては共通している．

しかし一方で，複数の単語から句や文の意味表現をベクトル空間上でどのように構成し，その類似度をどのように計算するかについては自明ではない．そのため，単語ベクトルを用いた句や文の意味表現の計算法や学習法が，一つの研究分野として新たに注目されるようになった [4], [17], [18]．この研究では具体的に，複数の単語ベクトルとそれらに対する構成関数 (行列やテンソル，ニューラルネットワークなどから作られる) を用いることで，句や文のベクトルを適切に表現し学習するモデルを構築する [5], [6], [7], [19]．これらの中でも特に重要なのが，Deep Learning の一手法である Recursive Neural Network (RNN) である．RNN は，構文木とニューラルネットワークを用いて，任意の長さの文を 1 つのベクトルで表現し学習することができる．まず，2 つの単語 w_i と w_j に対する d 次元ベクトル $\mathbf{d}(w_i)$ と $\mathbf{d}(w_j)$ から，句ベクトル $\mathbf{p}$ を以下の構成関数を用いて計算する．

$$\mathbf{p} = g\left(\mathbf{W}\begin{bmatrix}\mathbf{d}(w_i)\\ \mathbf{d}(w_j)\end{bmatrix}\right) = g(\mathbf{W}_l\mathbf{d}(w_i) + \mathbf{W}_r\mathbf{d}(w_j)).$$

ここで，$\mathbf{W} = [\mathbf{W}_l; \mathbf{W}_r] \in \mathbb{R}^{d\times 2d}$ は学習する重み行列 (; は行列の連結を表す)，g は tanh などの非線形関数である．次に，この $\mathbf{W}$ による線形変換と g による非線形変換を，文の木構造に合わせすべての単語ペアに対して再帰的に適用し，最終的に一つの文ベクトルを得る．ここでこの文ベクトルは，単語と同じ d 次元表現となることに注意されたい．

2.2 計量学習とその非線形拡張

本稿の冒頭で述べた類似度 (以降，より一般的に計量と呼ぶ) は，自然言語処理だけでなく機械学習全般において重要である．なぜならすべての機械学習手法は，各々のデータ点が適切な計量を持つことが前提となっており，それに基づき分類，クラスタリング，回帰などが行われる (ここでの適切な計量とは，似たデータであればベクトル空間においてそれらの距離が小さい，あるいは内積が大きいということを意味する)．そのため機械学習の分野では，計量学習と呼ばれる分野が古くから存在する [8]．適切な計量

とは解くべき問題に依存して決まるものの，どのような場合においても計量学習の最終的な目標は同じである．それは，問題を解く上でより適切な距離や類似度を持つベクトル空間を，新たに獲得することである．

例えば距離学習 [20], [21] では主に，以下のマハラノビス距離と呼ばれる計量を最適化する．

$$D_{\mathbf{M}}(\mathbf{x},\mathbf{x}') = (\mathbf{x}-\mathbf{x}')^{\mathrm{T}}\mathbf{M}(\mathbf{x}-\mathbf{x}').$$

ここで $\mathbf{x}$ と $\mathbf{x}'$ は特徴ベクトル，$\mathbf{M}$ は学習する変換行列 (半正定値行列) である．また一方で類似度学習 [22], [23] では主に，内積に基づく計量を学習する．

$$K_M(\mathbf{x},\mathbf{x}') = \mathbf{x}^{\mathrm{T}}\mathbf{M}\mathbf{x}'.$$

ここで，$\mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W}$ のように分解することで，式 (2) は以下のように書くことができる．

$$\begin{aligned} D_{\mathbf{M}}(\mathbf{x},\mathbf{x}') &= (\mathbf{x}-\mathbf{x}')^{\mathrm{T}}\mathbf{M}(\mathbf{x}-\mathbf{x}') \\ &= (\mathbf{x}-\mathbf{x}')^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{W}(\mathbf{x}-\mathbf{x}') \\ &= (\mathbf{W}\mathbf{x}-\mathbf{W}\mathbf{x}')^{\mathrm{T}}(\mathbf{W}\mathbf{x}-\mathbf{W}\mathbf{x}') \\ &= ||\mathbf{W}\mathbf{x}-\mathbf{W}\mathbf{x}'||^2. \end{aligned}$$

同様に式 (3) も最終的には以下のように書くことができる．

$$K_{\mathbf{M}}(\mathbf{x},\mathbf{x}') = (\mathbf{W}\mathbf{x})^{\mathrm{T}}(\mathbf{W}\mathbf{x}').$$

つまり，距離ないしは類似度学習は，元のデータ点 $\mathbf{x}$ と $\mathbf{x}'$ を適切なユークリッド距離ないしは内積を持つ空間へと変換する行列 $\mathbf{W}$ を学習することと等価である．

さらにこの計量学習は，カーネル関数を用いることで非線形に拡張することができる [9], [24], [25], [26]. カーネル関数を K とすると，元データ $\mathbf{x} \in \mathcal{X}$ を高次元再生核ヒルベルト空間 $\mathcal{H}$ へ，その高次元ベクトル表現 $\boldsymbol{\phi}(\mathbf{x})$ を陽に計算することなく，その内積 $\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}')$ のみを用いることで写像し表現することができる．我々は計量学習におけるカーネル化によって，問題を解くためのより良い高次元空間を新たに獲得する．

2.3 ペアデータに対するニューラルネットワーク

近年の Deep Learning の発展に伴い，ニューラルネットワークをペアデータに対して用いる研究が盛んに行われている．ペアデータとは例えば，異なる言語のパラレルコーパスや，画像データとその説明文のペアなどである．

Hermann と Blunsom [27] はパラレルコーパスを用いて，異なる言語で書かれているが内容が同一の文書をベクトルで表現し，それらのユークリッド距離を最小化することで，様々な言語の単語ベクトルを学習する手法を提案している．また Gao らも [28] パラレルコーパスを訓練データとして，機械翻訳のためのニューラルネットワークを構築している．このモデルでは，原言語と目標言語のペアを共通の低次元ベクトル空間で表現し，それらの翻訳スコアを距離や内積を用いて計算し最適化する．さらに Kiela と Bottou [29] は，画像と言語のペアデータ利用したマルチモーダルなニューラルネットワークを提案している．

このように，ペアデータを用いた研究は様々にあるものの，これらの研究を 2.2 節の計量学習の観点から捉え直すと結局は，$\mathbf{x}$ と $\mathbf{x}'$ を特徴ベクトル，N を非線形変換とした上で，距離 $||N(\mathbf{x}) - N(\mathbf{x}')||^2$ や内積 $N(\mathbf{x})^{\mathrm{T}}N(\mathbf{x}')$ をニューラルネットワークを用いて最適化していることと等価である．つまり，計量学習における変換行列 $\mathbf{W}$ に tanh などの非線形関数を適用した拡張であり，これは主に情報検索分野などで Deep Metric Learning と呼ばれている [30].

2.4 確率分布あるいは BOWs データのカーネル埋め込み

カーネル埋め込み (Kernel Embeddings) と呼ばれる手法では主に，確率分布が持つ高次モーメント情報をガウシアンカーネルなどを用いて保持した上で，それを再生核ヒルベルト空間 $\mathcal{H}$ 上の一点として表現する [31], [32]．これを画像処理や自然言語処理における分類タスクで用いる際には，画像や文書などのデータを分布によって表現した上で，その分布間のカーネルを計算し SVM を適用する [33], [34]．より具体的に文書分類タスクを例に挙げると，まず文書データを単語ベクトルの集合として表現し，それを $\mathbf{D}$ とする．次に，2 つの文書データ $\mathbf{D}$ と $\mathbf{D}'$ 間の類似度 $sim(\mathbf{D},\mathbf{D}')$ を，分布間のカーネル埋め込みを用いて以下のように計算する．

$$sim(\mathbf{D},\mathbf{D}') = \frac{1}{|\mathbf{D}||\mathbf{D}'|}\sum_{i=1}^{|\mathbf{D}|}\sum_{j=1}^{|\mathbf{D}'|}K(\mathbf{x}_i,\mathbf{x}_j).$$

ここで $|\mathbf{D}|$ は文書に含まれる単語数，$\mathbf{x}$ は単語のベクトルを表す．つまりこの文書間類似度は，2 つ文書に出現するすべての単語ベクトル間のカーネル平均を計算しているに過ぎない．このように，文書間の類似度を分布間のカーネル埋め込みとして定義できたことで，最終的にはこのカーネルに基づき通常の SVM を用いて文書を分類することができる．

これもまた，計量学習の観点から捉え直すと，ベクトルの集合として表現されたデータに対して適切な計量を学習していることに相当する．しかし，このような確率分布間に対するカーネル埋め込みでは，埋め込むデータ表現が Bog Of Words (BOWs) に限定されてしまう．そのため，より複雑な構造を持つデータに適用する際には，BOWs とは異なる表現とそれに対する新たなカーネル埋め込みのアプローチが必要となる．

3. 提案法

この章で我々は，ニューラルネットワークによる意味構成とそのカーネル埋め込みを用いた多層非線形類似度学

習を提案する．訓練データは，$\{((S_i, S'_i), y_i)\}_{i=1}^{N}$ の形式で与えられられる．ここで，(S_i, S'_i) は 2 つの文 S のペア，$y_i \in [0.0, 1.0]$ は (S_i, S'_i) に対する正規化された類似度スコアとする．

我々の最終目標は，文の意味構成のための新たな単語表現学習モデルを構築することである．そのために我々は，低次元空間における文の構造と意味の表現 (3.1 節) と，高次元空間におけるそのカーネル埋め込みを用いた非線形類似度学習 (3.2 節) とを組み合わせる．そして最終的に，カーネル埋め込みの多層化を試みる (3.4 節)．このような意味構成モデルに対する我々のアプローチは，以下の仮説に基づいている．

- 何らかの構成的な表現 (ここでは文とする) は，それを構成する要素 (この場合は個々の単語) が表現されている元空間よりも，表現力の高い別空間において学習する必要がある．

ここで我々は，ベクトル空間における意味構成のモデル化を，構成的表現の計量学習という観点から捉え直す．すると，ニューラルネットワークを用いた意味構成の従来研究が，低次元空間において詳細かつ複雑にモデル化する制約を受けた上で計量を学習していることに気づく (図 2 の上段を参照)．これを踏まえ我々は，カーネル関数によって写像される高次元空間が，構成に伴なって生じる複雑な構造と豊かな意味を，よりシンプルに表現し学習できることに着目する (図 2 の下段と 5.4 節参照)．

3.1 低次元空間での文の構造と意味の表現

この節では，文の構造と意味を表現するための，幾つかの構成関数を導入する．我々は，文 S の構造と意味を表現する低次元ベクトル $\mathbf{x}$ を，構成関数 f を用いて計算する．特に本稿では，以下の 2 つの関数を用いる．

まず最も単純に，文ベクトルを $\mathbf{x} = f_{SUM}(S) = \sum_{w \in S} \mathbf{d}(w)$ と計算する．ここで，w は文 S に含まれる単語，$\mathbf{d}(w) \in \mathbb{R}^d$ は単語 w に対する d 次元ベクトルである．これは単語ベクトルの単純な和に基づく文の BOWs 表現であり，文が持つ構造情報は保持されず，文に含まれる単語の共起情報のみが考慮される．

もう一つの構成関数として我々は，単語ベクトルに対する連結演算を用いて，文の構造を保持したベクトルを以下のように計算する．

$$\begin{aligned}\mathbf{x} &= f_{STR}(P_S) \\ &= g\left(\sum_{(w_i, w_j) \in P_S} h\left(\mathbf{W}\begin{bmatrix}\mathbf{d}(w_i) \\ \mathbf{d}(w_j)\end{bmatrix}\right)\right). \end{aligned} \tag{1}$$

ここで P_S は，文 S が持つ構文的あるいは意味的に関係のある単語ペアの集合，$\mathbf{W} \in \mathbb{R}^{d \times 2d}$ は学習する重み行列，w_i は文 S における i 番目の単語，そして g と h は線形関数あるいは非線形関数である tanh とする．特に，重み行列 $\mathbf{W}$ は以下の 2 通りで与えられる．

$$\mathbf{W} = \begin{cases}\mathbf{W}_{(pos(w_i), pos(w_j))} \\ \mathbf{W}_{(sem(w_i), sem(w_j))},\end{cases} \tag{2}$$

ここで，$pos(w)$ は単語 w に対する品詞タグ，$sem(w)$ は単語 w に対する述語項構造ラベル (例えば，意味上の主語や目的語などを表すタグ) である．このように，各単語ペアが持つ構造ラベルペアに対して重み行列を設定し学習することで，文が持つ構造の情報を適切に保持しつつ学習することが可能となる．さらに我々は，項を 2 つ持つ単語に対して，上式を以下のように拡張する．

$$\begin{aligned}\mathbf{x} &= f_{STR}(P_S) \\ &= g\left(\sum_{(w_i, w_j, w_k) \in P_S} h\left(\mathbf{W}\begin{bmatrix}\mathbf{d}(w_i) \\ \mathbf{d}(w_j) \\ \mathbf{d}(w_k)\end{bmatrix}\right)\right). \end{aligned} \tag{3}$$

この構成関数は，文の動詞における意味上の主語や目的語などの関係を，より自然かつ直接的に捉えることができる．例えば，*"A man in a blue jumpsuit is courageously performing a wheelie on a motorcycle."* における *"man-performing-wheelie"* や，*"A girl in a uniform, which is blue, is quickly raising her arm."* における *"girl-raising-arm"* などである．

上記の構成関数は，重み行列の設定に述語項構造ラベルを用いる点などを除けば，Socher ら [10] のモデルと類似している．またこれまで，テンソル演算やより多層のニューラルネットワークなどを用いる構成関数 [7] が様々に提案されているが，これらに関しては今後の課題とする．

3.2 カーネル埋め込みを用いた高次元空間における非線形類似度学習

この節で我々は，カーネル埋め込みを用いた非線形類似度学習を提案する．まず，ベースラインである線形カーネル K として，正規化された内積であるコサイン類似度 $K_{cos}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathrm{T}}\mathbf{x}'/||\mathbf{x}||||\mathbf{x}'||$ を用いる．ここで，$\mathbf{x}$ と $\mathbf{x}'$ は 3.1 節で計算した文ベクトルである．本稿では，適切な類似度計算のため，あるいは計算の単純化のためだけでなく，学習時にカーネルの値を制御するために，すべてのカーネル関数 K を以下のように正規化する．正規化されたカーネルは，写像された空間におけるコサイン類似度となり，以下のように表現される．

$$K_{cos}(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}')) = \frac{K(\mathbf{x}, \mathbf{x}')}{\sqrt{K(\mathbf{x}, \mathbf{x})}\sqrt{K(\mathbf{x}', \mathbf{x}')}}. \tag{4}$$

ここで $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{\phi}$ は，非線形カーネル関数 K によって写像された高次元空間における $\mathbf{x}$ の特徴ベクトルである．このような非線形カーネルとして我々は，以下の正規化された多項式カーネル K_{poly} とガウシアンカーネル K_{rbf} の 2 つを用いる．

$$K_{poly}(\mathbf{x},\mathbf{x}') = \left(\frac{c+K_{cos}(\mathbf{x},\mathbf{x}')}{c+1}\right)^p, \quad s.t\ \ c\geq 0,\ \ p\in\mathbb{N}, \tag{5}$$

$$K_{rbf}(\mathbf{x},\mathbf{x}') = \exp\left(\frac{K_{cos}(\mathbf{x},\mathbf{x}')-1}{\sigma^2}\right) \quad s.t\ \ \sigma\geq 0. \tag{6}$$

次に我々は，2 つの文 S と S' の類似度 $K(S,S')$ を，3.1 節の SUM と STR のモデルに対するカーネルを組み合わせて計算する．本稿では文の類似度計算に，以下の 2 つの関数を用いる．

$$K_{SUM}(S,S') = K(f_{SUM}(S), f_{SUM}(S')), \tag{7}$$

$$K_{SUM+STR}(S,S') = K(f_{SUM}(S), f_{SUM}(S')) + K(f_{STR}(P_S), f_{STR}(P_{S'})). \tag{8}$$

最終的な目的関数は以下の $L(\Theta)$ であり，訓練データ $\{((S_i,S_i'),y_i)\}_{i=1}^N$ に対してこれを最小化する．

$$L(\Theta) = \sum_{i=1}^{N}\frac{1}{2}\{y_i - K(S_i,S_i')\}^2 + \frac{\lambda}{2}||\Theta||^2. \tag{9}$$

ここで，Θ はモデルにおける学習パラメータの集合であり，単語ベクトル表現 $\mathbf{d}(w)$，重み行列 $\mathbf{W}$, そしてカーネル関数内のパラメータ (多項式カーネルでは c，ガウシアンカーネルでは σ) がこれに含まれる．このように我々は，文の高次元意味ベクトル表現を陽に計算することなく，カーネルによって写像された高次元空間においてその類似度のみを計算し学習することで，文の意味構成により適した単語表現を新たに獲得する (図 1 参照).

我々がこの提案法において特に強調したいのは，低次元空間における文の構造と意味の表現と，高次元空間におけるそのカーネル埋め込みを用いた非線形類似度学習とを組み合わせることである．これにより，低次元空間のみを用いて詳細かつ複雑にモデル化するニューラルネットワークよりも，意味構成のための単語表現学習をよりシンプルにデザインできる (図 2 と 5.4 節を参照)．また，本稿の主旨からは逸脱するが，ベクトルデータだけでなく構造データについても同様に，何らかのカーネルを定義した上で表現学習を適用することも可能である [*1].

3.3 学習における勾配計算

我々は，単語ベクトル $\mathbf{w}=\mathbf{d}(w)$ を学習するため，まず勾配 $\partial L/\partial\mathbf{w}$ を計算し，次に勾配法を用いて目的関数 L を最小化し，最終的に新たな単語ベクトル $\mathbf{w}_{new}$ を更新する．多項式カーネルとガウシアンカーネルを用いた場合，勾配は以下のように計算される．

[*1] 本稿では，文 S と S' は最終的にベクトルによって表現されるが，提案する表現学習法はグラフ構造や木構造を持つ構造データに対しても拡張できる [35].

$$\frac{\partial K_{poly}}{\partial\mathbf{w}} = p\left(\frac{c+K_{cos}}{c+1}\right)^{(p-1)}\left(\frac{1}{c+1}\right)\frac{\partial K_{cos}}{\partial\mathbf{w}},$$

$$\frac{\partial K_{rbf}}{\partial\mathbf{w}} = \exp\left(\frac{K_{cos}-1}{\sigma^2}\right)\left(\frac{1}{\sigma^2}\right)\frac{\partial K_{cos}}{\partial\mathbf{w}}.$$

ここで勾配 $\partial K_{cos}/\partial\mathbf{w}$ は，コサイン類似度の単語ベクトルによる微分で計算できる．コサイン類似度 $K_{cos}(\mathbf{x},\mathbf{x}') = \mathbf{x}^{\mathrm{T}}\mathbf{x}'/||\mathbf{x}||||\mathbf{x}'||$ の $\mathbf{x}$ による微分は，以下の通りである．

$$\frac{\partial K_{cos}(\mathbf{x},\mathbf{x}')}{\partial\mathbf{x}} = \frac{||\mathbf{x}||||\mathbf{x}'||\mathbf{x}'-\mathbf{x}^{\mathrm{T}}\mathbf{x}'||\mathbf{x}'||\frac{\mathbf{x}}{||\mathbf{x}||}}{(||\mathbf{x}||||\mathbf{x}'||)^2}.$$

この勾配を用いて，単語ベクトル $\mathbf{w}$ を以下のように更新する (ここで α は学習率).

$$\mathbf{w}_{new} = \mathbf{w} - \alpha\frac{\partial K}{\partial\mathbf{w}}$$

また我々は，重み行列に関しても勾配を計算し，単語ベクトルと同様に勾配法を用いて更新する．

$$\frac{\partial K_{poly}}{\partial\mathbf{W}} = p\left(\frac{c+K_{cos}}{c+1}\right)^{(p-1)}\left(\frac{1}{c+1}\right)\frac{\partial K_{cos}}{\partial\mathbf{W}},$$

$$\frac{\partial K_{rbf}}{\partial\mathbf{W}} = \exp\left(\frac{K_{cos}-1}{\sigma^2}\right)\left(\frac{1}{\sigma^2}\right)\frac{\partial K_{cos}}{\partial\mathbf{W}},$$

ここで $\partial K_{cos}/\partial\mathbf{W}$ は，コサイン類似度の重み行列による微分である．実際には，3.1 節の重み行列を用いた構成関数によって複雑に計算されるが，ここでは単純に，ベクトル $\mathbf{Wx}$ と $\mathbf{W}'\mathbf{x}'$ のコサイン類似度を例にしてこの勾配計算を述べると，以下のようになる．

$$\frac{\partial K_{cos}}{\partial\mathbf{W}} = \frac{||\mathbf{Wx}||||\mathbf{W}'\mathbf{x}'||(\mathbf{W}'\mathbf{x}')^{\mathrm{T}}\mathbf{x}-(\mathbf{Wx})^{\mathrm{T}}(\mathbf{W}'\mathbf{x}')\frac{(\mathbf{Wx})^{\mathrm{T}}\mathbf{x}}{||\mathbf{Wx}||}}{(||\mathbf{Wx}||||\mathbf{W}'\mathbf{x}'||)^2}.$$

さらに我々は，多項式カーネルとガウシアンカーネル内のパラメータである c と σ に関しても勾配を計算し，単語ベクトルと重み行列同様に勾配法を用いて更新する．

$$\frac{\partial K_{poly}}{\partial c} = p\left(\frac{c+K_{cos}}{c+1}\right)^{(p-1)}\left(\frac{1-K_{cos}}{(c+1)^2}\right)$$

$$\frac{\partial K_{rbf}}{\partial\sigma} = \exp\left(\frac{K_{cos}-1}{\sigma^2}\right)\left(\frac{2(1-K_{cos})}{\sigma^3}\right)$$

3.4 非線形類似度学習の多層化

提案法の最後として，カーネル埋め込みを多層化することを試みる．本稿で多層化するカーネルには，ガウシアンカーネルの他，以下の一般多項式カーネルを用いる．

$$K_{gpoly}(\mathbf{x},\mathbf{x}') = (1+K'_{cos}(\mathbf{x},\mathbf{x}'))^n \quad s.t\ \ |K'_{cos}(\mathbf{x},\mathbf{x}'))|<1,\ \ n\in\mathbb{R} \tag{10}$$

これは，多項式カーネルの次数 p を実数値 n としたものであり，本稿でこれを一般多項式カーネルと呼ぶ．これは，アイザック・ニュートンが 1665 年頃に発見した一般二項

定理を用いた，多項式カーネルの自然かつ容易な拡張である．式 (17) は，ガンマ関数

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$$

を用いて，以下のように無限級数展開される [*2]．

$$K_{gpoly}(\mathbf{x},\mathbf{x}') = \sum_{k=0}^{\infty} \frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)} K_{cos}(\mathbf{x},\mathbf{x}')^k.$$

この拡張により，次数 n も学習パラメータとすることで，多項式カーネルにおける素性の組み合わせ次数をも同時に最適化できる．

以上を踏まえて，ガウシアンカーネルと一般多項式カーネルを再帰的に適用して，各々を以下のように多層化する．

$$K^{\ell}_{rbf}(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{1-K^{\ell-1}_{rbf}(\mathbf{x},\mathbf{x}')}{\sigma^2_\ell}\right), \quad (11)$$

$$K^{\ell}_{gpoly}(\mathbf{x},\mathbf{x}') = (1+K^{\ell-1}_{gpoly}(\mathbf{x},\mathbf{x}'))^n. \quad (12)$$

ここで，ℓ は多層カーネルにおける層数であり，$\ell = 0$ の時を 1 層のガウシアンカーネル (式 (6)) と一般多項式カーネル (式 (10)) とする．このようにカーネルを多層化することで，カーネル内パラメータ (ガウシアンカーネルでは σ，一般多項式カーネルでは n) が増え，ℓ 次元ベクトルとなる．この多層カーネルを用いることで，表現力の高い高次元空間へ階層的に写像することができ，またそれは ℓ 次元のベクトルパラメータによって適切な類似度を持つ空間として制御される．目的関数は式 (9) と同様であり，学習は多層カーネルを再帰的に微分し 3.3 節で述べた勾配計算を用いて行う．

4. 実験

4.1 データセットと評価

提案法の評価には，自然言語処理のコンペティションである SemEval 2014 の文の意味的類似度データセット (Task 1: Sentences Involving Compositional Knowledge，以下 SICK) を用いる．この SICK は，与えられた 2 つの文に対する類似度を予測するタスクであり，その類似度は人手によって 1.0 から 5.0 の範囲でスコア付けされている (表 1 参照)．このデータセットには 9927 の文ペアが含まれており，訓練データ，開発データ，テストデータはそれぞれ 4500/500/4927 と分割されている．具体的な評価指標には，モデルによって計算された類似度とデータセットのスコアとのピアソン相関係数 r，スピアマン相関係数 ρ，平均二乗誤差 (MSE) を用いる．データセットとタスクに関する詳細は [36] を参照されたい．

[*2] この無限級数における係数は，n 次を過ぎた直後急激に減衰し，正と負の微小な値の両方を取るため，厳密にこれは正定値カーネルではないことに注意されたい．

4.2 実装の詳細

提案法における目的関数は式 (9) の $L(\Theta)$ であり，これを最小化する．Θ は提案法における学習パラメータの集合であり，ここでは以下の 3 つである．

(1) 訓練データに現れるすべての単語のベクトル．
(2) 構造ラベルペアを用いて設定した重み行列 $\mathbf{W}$(3.1 節参照)．
(3) 多項式カーネルとガウシアンカーネル内のパラメータである c と σ．

文の構文解析には Enju[*3] を用い，構文的あるいは意味的な関係にある単語間の品詞タグと述語項構造ラベルを用いて重み行列 $\mathbf{W}$ を設定する．

モデルのハイパーパラメータは，開発データを用いて調整する．まず我々は，単語ベクトル表現を以下の 4 つを用いて初期化する:

(1) Random ベクトル : ベクトルの各要素は，-1.0 から 1.0 の範囲でランダムな値に初期化する．
(2) LSA [15] : 潜在意味解析 (LSA) [1] によって得られた，50 次元の単語ベクトル表現．
(3) NLM [2][*4] : ニューラル言語モデル (Neural Language Model) によって学習された，50 次元の単語ベクトル表現．
(4) GloVe [3][*5] : 単語の共起頻度と最小二乗法に基づくモデルで学習された，50 次元の単語ベクトル表現．

また，すべての重み行列 $\mathbf{W}$ は，単位行列にガウシアンノイズを加えた $\mathbf{W} = \mathbf{I} + \epsilon$ で初期化する．さらに，カーネル内パラメータの初期値はそれぞれ，多項式カーネルでは $c = 1.0$，ガウシアンカーネルでは $\sigma = 1.0$ とする．特に多項式カーネルについては，すべての実験において次数 p を 4 とし，多層カーネルの場合もすべてのパラメータを同様に初期化し，3 層と 6 層の場合で実験を行う．モデルの学習は勾配法の一種である AdaGrad [37] を用いて行い，学習率はそれぞれ，単語ベクトル表現に対して $\alpha = 0.5$，重み行列 $\beta = 10^{-2}$，カーネル内パラメータ $\gamma = 10^{-3}$，そして正則化項については $\lambda = 10^{-6}$ とする．

4.3 比較する既存研究

まず我々は提案法と，コンペティションの上位チームである ECNU，The Meaning Factory，UNAL-NLP，Illinois-LH [38], [39], [40], [41] とを比較する．これらのチームはすべて，人手で設計した様々な種類の特徴量を用いている．また，自然言語処理における Deep Learning の代表的な手法である Recursive Neural Network (RNN) [10] と Long Short-Term Memory (LSTM) [11] とも比較する．これらのモデルではニューラルネットワークを用いて，文の意味ベクトル

[*3] http://www.nactem.ac.uk/enju/index.html
[*4] http://ronan.collobert.com/senna/
[*5] http://nlp.stanford.edu/projects/glove/

文 A	文 B	類似度スコア
A man and woman are talking.	A man and a woman are having a conversation.	4.8
A football player in a purple jersey is breathlessly running with the ball for a touchdown.	A football player in a purple jersey is running with the ball for a touchdown.	4.5
A woman is scrubbing a zucchini with a vegetable brush.	A woman is eating zucchini and vegetables and scrubbing with a brush.	3.2
Two people are carrying colorful baskets and blankets and walking near a building.	Two people are sitting with laden baskets and blankets.	2.8
A man is jumping into an empty pool.	There is no biker jumping in the air.	1.2
A person is not chopping an onion.	A person is riding a motorcycle.	1.1

表 1 SICK のデータセットに含まれる様々な文とその類似度スコアの一例.

Method	r	ρ	MSE
Cosine (SUM)	0.7588	0.7391	0.4820
Poly (SUM)	0.8332	0.7810	0.3205
RBF (SUM)	0.8339	0.7804	0.3162
Cosine (SUM + STR_POS)	0.7510	0.7429	0.4510
Poly (SUM + STR_POS)	0.8301	0.7858	0.3176
RBF (SUM + STR_POS)	0.8325	0.7721	0.3094
Cosine (SUM + STR_SEM)	0.7647	0.7435	0.4787
Poly (SUM + STR_SEM)	**0.8480**	**0.7968**	**0.2904**
RBF (SUM + STR_SEM)	0.8447	0.7923	0.2968

表 2 様々なモデルの相関係数と MSE の結果．ここでは，構成関数 f_{STR} における g と h をすべて線形としている．

を低次元空間のみを用いて表現し学習する．これらの手法の詳細については，5.4 節と 6 章の関連研究で述べる．

5. 結果と考察

5.1 類似度学習における線形と非線形モデルの比較

表 2 は，様々な構成関数 (ここで STR における g と h は線形とする) とカーネル関数 (コサイン，多項式カーネル，ガウシアンカーネル) を用いた時の，相関係数と MSE である．ここで用いた単語ベクトル表現はすべて，50 次元の GloVe である．考察は以下の通りである．

(1) 相関係数と MSE のすべてにおいて，非線形カーネル (多項式カーネルとガウシアンカーネル) を用いた時が，線形カーネル (コサイン) の結果を大きく上回っている．特に，多項式カーネル (SUM+STR_SEM) が，本稿で実験したモデルの中で最も良い性能を示しており，ピアソン相関係数が 0.8480, スピアマン相関係数が 0.7968, MSE が 0.2904 となっている．これらの結果は，高次元空間における非線形類似度学習が，文の意味構成により適した単語ベクトル表現の獲得に効果的であることを示している．

(2) STR モデル，つまり文の構造を考慮する方が，SUM モデル，つまり BOWs よりも全体的に高い性能を示している．特に，述語項構造解析のラベルを用いて重み行列を設定し学習するモデルである STR_SEM が，最も良い結果となっている．しかし一方で，非線形類似度学習を用いさえれば，たとえ SUM であったとしても 0.8 以上の高い相関係数を示していることが確認できる．この結果は，文を単語ベクトルの和で単純に表現したとしても，非線形類似度学習によって良い性能を達成できること，そしてこれは，ベクトルの和によって文における単語の共起情報を捉えることが，非常に重要だということを示唆している．

	r (g and h are identity)	r (g and h are $\tanh$)
Cosine	0.7647	0.7717
Poly	**0.8480**	0.8392
RBF	0.8447	0.8393

表 3 意味構成における線形と非線形モデルの比較．線形の意味構成と非線形の類似度学習の組み合わせが，高い相関係数を示している．

5.2 意味構成における線形と非線形モデルの比較

表 3 は，構成関数と類似度学習における線形と非線形の組み合わせパターンと，その相関係数 (ピアソン) の結果を示している．ここで用いた単語ベクトル表現はすべて，50 次元の GloVe であり，意味構成モデルは SUM+STR_SEM である．考察は以下の通りである．

(1) 非線形カーネル埋め込みを用いる方が，ニューラルネットワークの非線形関数を用いるよりも，相関係数の上昇が高いことが確認できる．これは，単語ベクトル空間において文の意味構成をモデル化する際は，低次元空間ではなく高次元空間を用いる方がより効果的であることを示している．

(2) 線形の意味構成と非線形の類似度学習の組み合わせが，この中で最も良い結果となっている．このモデルの組み合わせの大きな利点は，実装やその最適化が容

	r (RBF カーネル)	r (一般多項式カーネル)
1 層	0.8339	0.8332
3 層	0.8001	0.8276
6 層	0.7786	0.8187

表 4 多層カーネルを用いた際の結果．多層にすればするほど，相関係数が低下するのが確認できる．

易なことである．我々は，低次元空間で詳細かつ複雑にモデル化したニューラルネットワークを構築することなしに，低次元空間と高次元空間の双方を用いてよりシンプルに単語表現学習モデルを構築することができる．

5.3 カーネル埋め込みの多層化

表 4 は，多層カーネルを用いた際の相関係数の結果である．これを見る通り，カーネルの多層化によって相関係数が徐々に下がるのが確認できる．特に，ガウシアンカーネルの多層化では，一般多項式の多層化に比べてより相関係数の降下が大きい．これは，ガウシアンカーネルでは無限次元空間への写像となり，モデルが過学習しているためと考えられる．いずれにせよ，カーネルの単純な多層化とその学習のみでは性能の向上は期待できず，過学習を防ぐような最適化における何らかの別の戦略を取る必要がある．

5.4 既存研究との比較

表 5 は，様々な既存研究の結果と提案法の結果との比較を示している．考察は以下の通りである．

(1) 我々の提案法は，コンペティションの上位 4 チームを上回る結果を示している．これらの上位チームはすべて，人手で特徴量を設計して学習する素性エンジニアリングベースの手法である．このことから，結果だけでなく手法のシンプルさにおいても，提案法に優位性がある．提案法は主に，単語ベクトル表現の学習がベースとなる手法であり，詳細な素性の設計や外部リソースを一切必要としない．

(2) 提案法は RNN モデルである DT-RNN と SDT-RNN [10] よりも高い相関係数を示している．これらは，ニューラルネットワークの非線形関数を用いて低次元の文ベクトルを計算するモデルであり，類似度学習に非線形性を持たせる構造にはなっていない．また提案法は，たとえ構造を無視した BOWs モデルである SUM であったとしても RNN を上回っており，RNN は文の構造の詳細なモデル化によって，逆に性能を低下させる危険性があることを示している．一方で，提案法のカーネルを用いた非線形類似度学習は，より頑健である．

(3) 提案法は，LSTM の様々なモデルとほぼ同性能を達成しており，特にスピアマン相関係数では 2 位にランクしている．このこともまた，提案法の頑健さと高次元空間における類似度学習が効果的であることを示している．また図 2 は，ニューラルネットワークを用いて文ベクトル表現を直接計算する RNN や LSTM などの手法と，我々のカーネルを用いた提案法のモデルの比較図である．これらのモデルでは，ベクトル空間において意味構成をモデル化する際に，非線形性と高次元性において重要な類似点と相違点がある．RNN や LSTM ような文ベクトルを陽に計算するモデルでは，構文の構造とニューラルネットワークの構造の双方からくるモデル上の制約によって，単語と同一次元の空間において文を表現する必要が生じる．これとは対照的にカーネルを用いた提案法は，そのような制約を受けることなく，構造と意味をより柔軟に表現し適切に学習することができる．

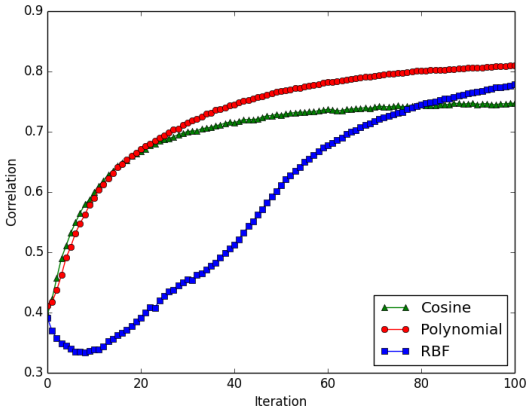

図 3 様々なカーネルを用いた時の学習曲線．横軸が訓練データに対するイテレーション回数，縦軸がテストデータに対する相関係数となっている．

5.5 カーネル内パラメータの最適化と学習曲線

図 3 は，様々な種類のカーネルを用いた時の，訓練データに対するイテレーション回数とテストデータに対する相関係数の挙動を示している．ここでは，多項式カーネルとガウシアンカーネル内のパラメータの初期値をそれぞれ $c = 5.0$，$\sigma = 5.0$ としている．

ここで興味深いのは，ガウシアンカーネルを用いた場合，訓練の初期段階では相関係数がコサインよりも低く，また相関係数も一度下がっていることである．しかし，ガウシアンカーネル内のパラメータである σ が学習されるに従い，相関係数は急激に上昇しコサインを上回り，最終的には多項式カーネルとほぼ同程度の結果となる．これは，ガウシアンカーネルを用いた非線形類似度学習の性能が，σ に大きく依存することを示している．このことは，σ が多

Method	r	ρ	MSE
Illinois-LH_run1 [41]	0.7993	0.7538	0.3692
UNAL-NLP_run1 [40]	0.8043	0.7458	0.3593
Meaning_Factory_run1 [39]	0.8268	0.7722	0.3224
ECNU_run1 [38]	0.8280	0.7689	0.3250
DT-RNN [10]	0.7863	0.7305	0.3983
SDT-RNN [10]	0.7886	0.7280	0.3859
LSTM	0.8477	0.7921	0.2949
Bidirectional LSTM [42]	**0.8522** (2)	**0.7952** (3)	**0.2850** (2)
2-layer LSTM [42]	0.8411	0.7849	0.2980
2-layer Bidirectional LSTM [42]	0.8488	0.7926	0.2893
Constituency Tree LSTM [11]	**0.8491** (3)	0.7873	**0.2852** (3)
Dependency Tree LSTM [11]	**0.8627** (1)	**0.8032** (1)	**0.2635** (1)
Our best model	0.8480	**0.7968** (2)	0.2904

表 5　既存研究との比較．(1)，(2)，(3) はそれぞれランキングを表す．

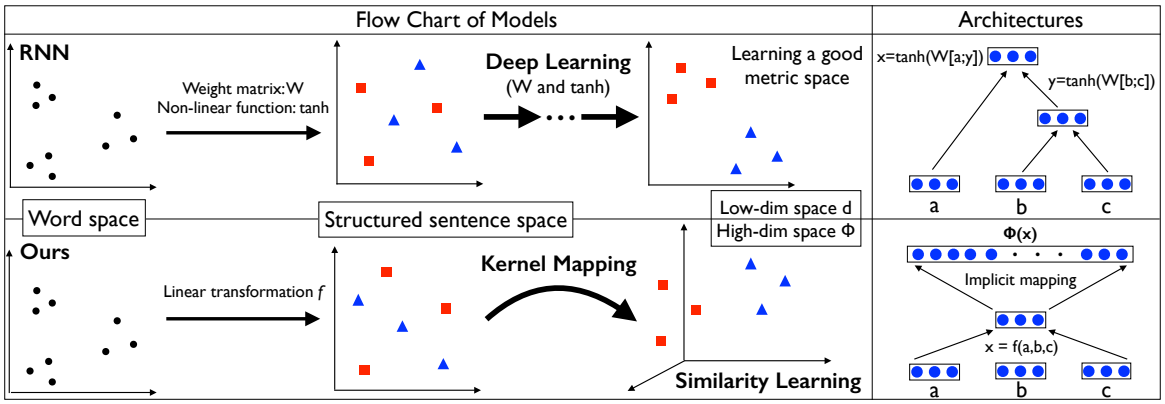

図 2　ベクトル空間における意味構成のモデル化を，計量学習という観点から捉え直した上で，RNN などのニューラルネットワークを用いた既存手法と，高次元カーネル空間を用いた提案法との違いを示す．

項式カーネルにおける c とは異なり，写像する空間の性質を大きく決定するパラメータであるためだと考えられる．

5.6　単語ベクトル表現の違いによる学習曲線の違い

図 4 は，様々な種類の単語ベクトル表現を用いた時の，訓練データに対するイテレーション回数と，テストデータに対する相関係数の挙動を示している．ここにおいては興味深い点が多くあり，それらを以下にまとめる．

(1) 初期値として相関係数が高いのが LSA と NLM であり，それぞれ約 0.6 である．これらの単語ベクトル表現は，大規模なコーパスを用いて学習されたものであるため，提案法を用いた学習なしでも文の意味的類似度をある程度適切に計算することができる．しかし，訓練段階になるとこれらは非常に対照的な挙動を示す．NLM では相関係数の上昇が顕著であるが，LSA のそれは非常に遅い．

(2) ランダムベクトルを初期値として用いる場合，学習前の相関係数は当然低い．また，LSA と NLM とは対照的に，大規模なコーパスで学習済みの GloVe の初期値も，ランダムベクトルと同様に低い相関係数となっている．しかし，ランダムベクトルと GloVe は訓練によって相関係数が急激に上昇する．特に GloVe は最終的に，4 種類のベクトルの中で最も良い相関係数を示している．またランダムベクトルは，LSA よりも高い相関係数を示している．

(3) これら結果は，連続値のベクトルであればたとえランダムベクトルであったとしても，学習が適切に進むことを示している．一方で LSA は，単語の共起頻度行列という離散的な特徴量から学習されており，このことが学習挙動の大きな違いの要因になっていると考えられる．またこれらの事実は，単語ベクトルを大規模コーパスを用いて事前に学習することよりもむしろ，

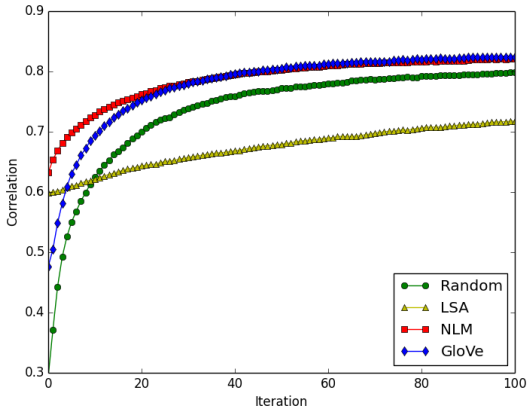

図 4 4 種類の単語ベクトルを用いた時の学習曲線．横軸が訓練データに対するイテレーション回数，縦軸がテストデータに対する相関係数となっている．

タスクに合わせた単語ベクトルを学習することの方がより重要であることを示している．

6. 関連研究

SemEval 2014 の SICK において成功したアプローチの多くは，素性エンジニアリングをベースとした手法を用いている [38], [39], [40], [41]．それらの特徴量は，自然言語処理でよく用いられる表層素性から，WordNet や Paraphrase Database [43] などの外部資源を使うものまで様々である [36].

一方で，単語ベクトル空間における文の意味構成モデルの研究では，自然言語処理における Deep Learning の一手法である Recursive Neural Network (RNN) [44], [45] が盛んに研究されている．このモデルでは，単語ベクトルを構文木に合わせて構成させ文ベクトルを計算し学習する．この RNN は，様々なタスクにおいて成功を収めているが，詳細に文の構造をニューラルネットワークを用いてモデル化しなければならないのに加え [6], [7]，ベースラインの BOWs(たとえば単語ベクトルの和) よりも良い性能を示すとは限らないことが近年の研究でわかってきた [10]．これは RNN が，複雑な構造を複雑なニューラルネットワークで直接モデル化することが原因であると考えられる．そのため，文の階層的な木構造を詳細に考慮するのではなく，単語間の関係を直接用いた Dependency Tree RNN (DT-RNN) [10] が提案された．その一方で，もうひとつのニューラルネットワークのリカレントニューラルネットワークのモデルである Long Short-Term Memory (LSTM) も，大きな成功を収めている [42], [46]．LSTM は，文の系列情報を保持するようなニューラルネットワークであり，構文解析の結果に依存することなく，より頑健に文の意味表現をモデル化できる．さらに近年，LSTM の系列情報に加えて，木構造や依存構造などを考慮した構造化された LSTM も提案されている [11].

7. 結論と今後の課題

本稿で我々は，ニューラルネットワークによる意味構成とそのカーネル埋め込みを用いた多層非線形類似度学習を提案した．提案法は，カーネル関数によって写像された高次元空間における文の類似度学習を通じて，意味構成のための単語表現を新たに獲得することを可能にする．我々はこの提案法を，文の意味的類似度データセットを用いて評価し，ベースラインである線形手法，Recursive Neural Network (RNN)，そしてコンペティションでの上位チームをすべて上回り，さらに構造化された Long Short-Term Memory (LSTM) と同程度の性能を達成した．またこのアプローチは、自然言語処理における Deep Learning の手法 [10], [11] のように，文の構造と意味を低次元空間において詳細かつ複雑にモデル化することなく，低次元空間と高次元空間の双方を用いることでそれらをよりシンプルに表現し学習することができる．

最後に今後の課題として，以下の 3 点を挙げる．

(1) 低次元空間における構造と意味の情報表現と，高次元空間における情報埋め込みと学習を，より統一的に行う枠組みを構築する．
(2) 低次の表現の組み合わせから高次の表現を構成する目的において，ニューラルネットワークとカーネルという 2 つの機械学習手法が持つ非線形性の役割を，より詳細に考察する．
(3) 近年の Deep Learning の発展に大きく貢献した Pre-training [47] や Drop-out [48] のような過学習を防ぐ手法を，多層非線形類似度学習においても新たに考案する．

参考文献

[1] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A.: Indexing by latent semantic analysis, *JASIS* (1990).
[2] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P.: Natural Language Processing (Almost) from Scratch, *The Journal of Machine Learning Research (JMLR)* (2011).
[3] Pennington, J., Socher, R. and Manning, C. D.: Glove: Global vectors for word representation, *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)* (2014).
[4] Mitchell, J. and Lapata, M.: Composition in Distributional Models of Semantics, *Cognitive Science*, Vol. 34, No. 8, pp. 1388–1439 (2010).
[5] Tsubaki, M., Duh, K., Shimbo, M. and Matsumoto, Y.: Modeling and Learning Semantic Co-Compositionality through Prototype Projections and Neural Networks, *Proceedings of the Conference on Empirical Methods on Natural Language*

Processing (EMNLP) (2013).

[6] Socher, R., Huval, B., Manning, C. D. and Ng, A. Y.: Semantic Compositionality through Recursive Matrix-Vector Spaces, *EMNLP-CoNLL* (2012).

[7] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y. and Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank, *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)* (2013).

[8] Bellet, A., Habrard, A. and Sebban, M.: A Survey on Metric Learning for Feature Vectors and Structured Data, *CoRR*, Vol. abs/1306.6709 (2013).

[9] Kedem, D., Tyree, S., Sha, F., Lanckriet, G. R. and Weinberger, K. Q.: Non-linear metric learning, *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)* (2012).

[10] Socher, R., Le, Q. V., Manning, C. D. and Ng, A. Y.: Grounded Compositional Semantics for Finding and Describing Images with Sentences, *Transactions of the Association for Computational Linguistics (TACL)* (2014).

[11] Tai, K. S., Socher, R. and Manning, C. D.: Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks, *arXiv preprint arXiv:1503.00075* (2015).

[12] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D. and Lai, J. C.: Class-based n-gram models of natural language, *Computational linguistics*, Vol. 18, No. 4, pp. 467–479 (1992).

[13] Blei, D. M., Ng, A. Y. and Jordan, M. I.: Latent dirichlet allocation, *the Journal of machine Learning research*, Vol. 3, pp. 993–1022 (2003).

[14] Widdows, D. and Cohen, T.: The semantic vectors package: New algorithms and public tools for distributional semantics, *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, IEEE (2010).

[15] Turney, P. D.: Domain and function: A dual-space model of semantic relations and compositions, *Journal of Artificial Intelligence Research (JAIR)* (2012).

[16] Bengio, Y., Ducharme, R., Vincent, P. and Janvin, C.: A neural probabilistic language model, *Journal of Machine Learning Research (JMLR)* (2003).

[17] Erk, K.: Vector Space Models of Word Meaning and Phrase Meaning: A Survey, *Language and Linguistics Compass*, Vol. 6, No. 10, pp. 635–653 (2012).

[18] Baroni, M., Bernardi, R. and Zamparelli, R.: Frege in space: A Program for Compositional Distributional Semantics, *Linguistic Issues in Language Technologies* (2013).

[19] Van de Cruys, T., Poibeau, T. and Korhonen, A.: A Tensor-based Factorization Model of Semantic Compositionality, *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)* (2013).

[20] Xing, E. P., Jordan, M. I., Russell, S. and Ng, A. Y.: Distance metric learning with application to clustering with side-information, *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)* (2002).

[21] Weinberger, K. Q., Blitzer, J. and Saul, L. K.: Distance metric learning for large margin nearest neighbor classification, *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)* (2005).

[22] Qamar, A. M., Gaussier, E., Chevallet, J.-P. and Lim, J. H.: Similarity learning for nearest neighbor classification, *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (2008).

[23] Chechik, G., Sharma, V., Shalit, U. and Bengio, S.: Large scale online learning of image similarity through ranking, *The Journal of Machine Learning Research (JMLR)* (2010).

[24] Chatpatanasiri, R., Korsrilabutr, T., Tangchanachaianan, P. and Kijsirikul, B.: On Kernelizing Mahalanobis Distance Learning Algorithms, *Arxiv preprint. http://arxiv. org/abs* (2008).

[25] Chatpatanasiri, R., Korsrilabutr, T., Tangchanachaianan, P. and Kijsirikul, B.: A new kernelization framework for Mahalanobis distance learning algorithms, *Neurocomputing*, Vol. 73, No. 10, pp. 1570–1579 (2010).

[26] Jain, P., Kulis, B., Davis, J. V. and Dhillon, I. S.: Metric and kernel learning using a linear transformation, *The Journal of Machine Learning Research (JMLR)*, Vol. 13, No. 1, pp. 519–547 (2012).

[27] Hermann, K. M. and Blunsom, P.: Multilingual Models for Compositional Distributed Semantics, *Proceedings of the Conference on Association for Computational Linguistics (ACL)* (2014).

[28] Gao, J., He, X., Yih, W.-t. and Deng, L.: Learning Continuous Phrase Representations for Translation Modeling, *Proceedings of the Conference on Association for Computational Linguistics (ACL)* (2014).

[29] Kiela, D. and Bottou, L.: Learning Image Embeddings using Convolutional Neural Networks for Improved Multi-Modal Semantics, *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)* (2014).

[30] Wu, P., Hoi, S. C., Xia, H., Zhao, P., Wang, D. and Miao, C.: Online multimodal deep similarity learning with application to image retrieval, *Proceedings of the ACM international conference on Multimedia* (2013).

[31] Kanagawa, M. and Fukumizu, K.: Recovering distributions from Gaussian RKHS embeddings, *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)* (2014).

[32] Kanagawa, M., Nishiyama, Y., Gretton, A. and Fukumizu, K.: Monte Carlo Filtering Using Kernel Embedding of Distributions, *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)* (2014).

[33] Muandet, K., Fukumizu, K., Dinuzzo, F. and Schölkopf, B.: Learning from distributions via support measure machines, *Advances in neural information processing systems (NIPS)*.

[34] Yoshikawa, Y., Iwata, T. and Sawada, H.: Latent support measure machines for bag-of-words data classification, *Advances in Neural Information Processing Systems (NIPS)* (2014).

[35] Srivastava, S., Hovy, D. and Hovy, E.: A Walk-Based Semantically Enriched Tree Kernel Over Distributed Word Representations, *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)* (2013).

[36] Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S. and Zamparelli, R.: SemEval-2014 Task 1: Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Textual Entailment, *SemEval* (2014).

[37] Duchi, J., Hazan, E. and Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization, *JMLR* (2011).

[38] Zhao, J., Zhu, T. T. and Lan, M.: ECNU: One Stone Two Birds: Ensemble of Heterogenous Measures for Semantic Relatedness and Textual Entailment, *SemEval* (2014).

[39] Bjerva, J., Bos, J., van der Goot, R. and Nissim, M.: The Meaning Factory: Formal Semantics for Recognizing Textual Entailment and Determining Semantic Similarity, *SemEval* (2014).

[40] Jimenez, S., Duenas, G., Baquero, J., Gelbukh, A., Bátiz, A.

J. D. and Mendizábal, A.: UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment, *SemEval* (2014).

[41] Lai, A. and Hockenmaier, J.: Illinois-lh: A denotational and distributional approach to semantics, *SemEval* (2014).

[42] Graves, A., Jaitly, N. and Mohamed, A.-R.: Hybrid speech recognition with deep bidirectional LSTM, *Automatic Speech Recognition and Understanding (ASRU)* (2013).

[43] Ganitkevitch, J., Van Durme, B. and Callison-Burch, C.: PPDB: The Paraphrase Database., *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies(NAACL)* (2013).

[44] Goller, C. and Kuchler, A.: Learning task-dependent distributed representations by backpropagation through structure, *Neural Networks*, Vol. 1, pp. 347–352 (1996).

[45] Socher, R., Lin, C. C., Manning, C. and Ng, A. Y.: Parsing natural scenes and natural language with recursive neural networks, *Proceedings of the International Conference on Machine Learning (ICML)* (2011).

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

[47] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P. and Bengio, S.: Why does unsupervised pre-training help deep learning?, *The Journal of Machine Learning Research (JMLR)*, Vol. 11, pp. 625–660 (2010).

[48] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research (JMLR)*, Vol. 15, No. 1, pp. 1929–1958 (2014).