

表形式からの分散表現獲得

吉田 稔^{1,a)} 松本 和幸^{1,b)} 北 研二^{1,c)}

概要: 表形式における単語埋め込み手法について考える。通常の文章を対象とした学習手法に比べ、表形式においては、単語の周辺文脈として、縦方向・横方向の二種類の文脈が考えられる。本稿では、これら同時に考慮した、二次元的なモデルを提案し、一次元的なモデルとの比較を行う。

1. はじめに

ニューラル単語埋め込み（単語分散表現）は、単語の意味を実数ベクトルに変換する効果的な手法として、近年盛んに研究が行われており、獲得されたベクトル表現は、単語類似度のみならず、ベクトル同士の演算による単語の類推にも威力を発揮することが知られている。([9]).

通常、文章を対象とした単語分散表現の獲得においては、文章を単語のリストと捉え、各単語と周辺単語の関係を学習するモデルが採用される。これに対し、本稿では、表形式に着目し、表形式中に現れる単語に対して分散表現を獲得する手法について論じる。表形式の一番大きな特徴は、二次元的であるということである。すなわち、表形式中の各セルは、同一行（横方向）、同一列（縦方向）の二つの方向に関連するセルを持ち、このどちらもが文脈として重要であると考えられる。例えば、表1の例では、“25”が“鈴木一郎”に関連している（“鈴木一郎”は25歳である）のみならず、“age”とも関連している（“25”は属性“Age”の属性値である）。さらに、同一列のセル“25”、“30”、“32”も、同じ属性の値であるという意味で関連している。

すなわち、表形式のための分散表現学習においては、通常の一次元的モデルではない、新しいモデルが必要となることが予想される。我々は、表形式中の単語の分散表現学習に適した、表形式からの分散表現を新たに提案する。

表 1 表形式の例

名前	年齢
鈴木一郎	25
山田太郎	30
田中花子	32

¹ 徳島大学大学院理工学研究部
Tokushima University

a) mino@is.tokushima-u.ac.jp

b) matumoto@is.tokushima-u.ac.jp

c) kita@is.tokushima-u.ac.jp

本研究では、特に、表形式中の属性に焦点を当てる。属性の分散表現において問題となるのは、共起文脈と属性文脈という、性質の異なる二種類の文脈が考えられることである。ここで共起文脈とは、共起する属性をもって文脈とする考え（例えば、“CPU”の文脈として、同一表で共起する属性“RAM”がとれる）のことで、属性文脈とは、とりうる属性値をもって文脈とする考え（例えば、“誕生日”の文脈として、日付表現がとれ、これに基づく類義語として“開催日”がとれる）のこととする。これは、通常の文章を対象とした、「周辺単語文脈」「係り受け文脈」「文書番号文脈」といった文脈の違いとは性質の異なるものであり、別の取り扱いが必要であると考えられる。このため、実験では、単語類似度タスクのほかに、「属性推測タスク」を新たに提案し、精度測定の指標とする。

2. 関連研究

文章からのニューラル単語埋め込み獲得については、word2vec ([9]) や GloVe ([12]) を端緒とし、数多くの研究が行われている。

また、人手による知識ベース ([16] [8] [2]) からの分散表現学習や、それらを、文章からの分散表現学習に組み込む手法 ([11] [15] [1]) についても、盛んに研究が行われている。これに対し、表形式については、知識ベースほど構造が明確化されておらず、新たな取り扱いが必要になると考えられる。

表形式を知識獲得に用いる研究は、利用できるテキストデータの増大に伴い、近年盛んになっており、大量のWWW表形式から属性知識を獲得する研究 [3]、表形式の複数列を利用した質問応答システム [13]、知識ベースのラベルを表形式に付与する研究 [7]、Wikipedia表形式からタプル表現を獲得する研究 [10] 等、多様な研究が報告されている。本稿で提案する表形式単語埋め込みを利用することにより、表形式の意味表現をより柔軟に行うことができ、

これら表形式を利用した知的システムの精度向上に役立つことが期待できる。

3. 使用データおよび記法の定義

日本語 Wikipedia の 2013 年版をデータとして用いる。^{*1} セルの集合を C とし、各セル $c \in C$ は、単語 $w(c)$ で代表されるとする。また、 c は、位置 (x_c, y_c) を持つ。ここで、 x_c は横方向の位置（列番号）、 y_c は縦方向の位置（行番号）であるとする。

本研究では、第一行目のセルは属性を表現していると仮定し、他のセルとは異なる前処理を行う。属性の位置判定については過去に様々な研究がある ([5][17]) が、これらの処理はエラーの混入が避けられないため、入力の実定性を考慮し、第一行目を属性と決め打ちする戦略をとる。なお、データ中の表形式^{*2}が row-wise（同一エンティティを同一行に並べ、同一属性の値を同一列に並べる）であった。

第一行目のセルを属性セル、その他のセルを属性値セルと呼ぶ。^{*3} 属性セルにおいては、セル中の文字列全体をそのまま単語として抽出する。これに対し、属性値セルに対しては、形態素解析^{*4}を行い、最後の単語のみを head として抽出する。表形式のセルは、多くの場合名詞句であり、最後の単語を抽出することにより、多くの場合で意味を適切に抽象化できると考えられる。

なお、数値表現については、有効数字を 2 桁として、それより先の桁を 0 として正規化した数値を、単語として用いる。（例えば、“123,000” という数字表現は、“120000” として扱われる。）

各セル $c \in C$ は、水平文脈 H_c (c と同一行のセルの単語集合)、および垂直文脈 V_c (c と同一列のセルの単語集合) という二種類の文脈を持つとする。左・右の一次元的文脈を考慮する通常の単語分散表現学習に比べ、表形式の単語分散表現では、これら二種類の文脈について考慮する必要がある。

4. 提案手法

提案手法では、word2vec と同様、Negative Sampling を利用した Skip-gram モデル (SGNS) を用いる。このため、本節では、まず SGNS について紹介し、その後その拡張として提案手法について述べる。いずれの手法でも、目的関数は、各対象セル（対象語）とその文脈セル（文脈語）の間で定義されるスコアの和として定義される。

*1 表形式の数は 255,039 であった。

*2 284 個中 266 個 (93.7%)

*3 属性そのものを持たない表も中には存在するが、そのような表の第一行目を属性として扱う。通常、属性値は属性に比べ頻度が低いので、モデル全体への影響は少ないと考えられる。

*4 本研究においては、Kuromoji を使用した。

4.1 Skip-gram model with Negative Sampling (SGNS)

SGNS のスコアは、訓練データにおける、“input word” w_I からの “output word” w_O の予測確率をもとに定義される。word2vec の実装においては、 w_I が対象語、 w_O が文脈語となっている。各 (w_I, w_O) のペアに対するスコアは、以下で定義される。

$$\log \sigma(v'_{w_O} \cdot v_{w_I}) + \sum_{k=1}^K \log \sigma(-v'_{w_k} \cdot v_{w_I})$$

ここで、 K は Negative sample の数、 v_w と v'_w はそれぞれ、「入力ベクトル」「出力ベクトル」と呼ばれる二種類のベクトルであり、 σ はシグモイド関数である。Negative sample w_k は、各学習ステップにおいてランダムサンプリングされる。

SGNS では、この関数を最大化することにより分散表現ベクトル v_w と v'_w を得る。

4.2 表形式からの分散表現の獲得

提案手法は、上記 SGNS を、表形式に適用できるように拡張したものとして定義できる。

SGNS では「入力ベクトル」「出力ベクトル」という二種のベクトルを用いていたが、これに対し、提案手法では、「属性ベクトル」 v_w^a 、「属性文脈ベクトル」 v_w^{ac} 、「属性値ベクトル」 v_w^v 、「属性値文脈ベクトル」 v_w^{vc} の四種のベクトルを用いる。

以下に、これら四種のベクトルについて説明する。（図 1 に、これらの関係を図示する。）なお、以下の説明では、 w を対象語、 z をその文脈語とする。

属性ベクトル・属性文脈ベクトル: 属性ベクトルと属性文脈ベクトルは、属性セル（第一行目のセル）に属する語を表現するために用いられる。これらの関係は、SGNS における入力ベクトルと出力ベクトルと同様、「属性ベクトル」で表現された属性語を、「属性文脈ベクトル」で表現された周辺の属性語から推定する、という仮定に基づき、 v_w^a と v_z^{ac} の内積によりスコアを定義する。

属性値ベクトル・属性値文脈ベクトル: 上記と同様、属性値ベクトル v_w^v で表現された各語を、属性文脈ベクトル v_z^{vc} で表現された文脈語から推定するというモデルに基づき、 v_w^v と v_z^{vc} の内積としてスコアを定義する。

上記の説明では、水平方向の文脈語（同一行にある文脈語との関係）についてのみ述べたが、提案手法では、これに加え、垂直方向の文脈関係（同一列にある文脈語との関係）についても考慮する。具体的には、 v_w^a と v_z^{vc} の内積および、 v_w^{ac} と v_z^{vc} の内積としてスコアが定義される。これを上記の水平文脈によるスコアと組み合わせることで、表 1 のような垂直方向の関係・水平方向の関係を同時にモデ

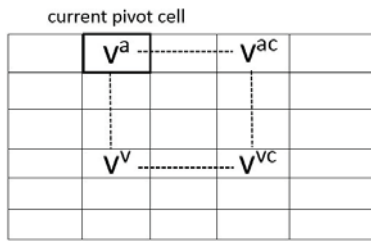


図 1 ベクトル間の関連のイメージ

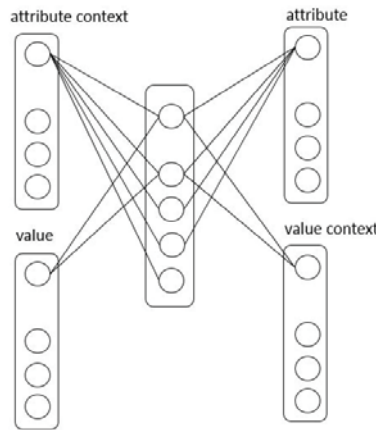


図 2 提案モデルのグラフ表現

ル化することができる。

図 2 に、提案モデルのグラフ表現を示す。SGNS と同様、各セルにおいて、対象語 w と文脈語 c のペアを一つずつとり、 w と c それぞれについて、四種のベクトルから一種が選択され、その間の重みを、スコアを増大させる方向に学習する。

4.3 目的関数

モデルの効果を測定するため、段階的に発展させた下記の四種のモデルを用いる。それぞれのモデルは、上で述べたベクトル同士の関係のうちいくつか（あるいは全て）を用いる。

4.3.1 Horizontal

Horizontal（水平目的関数）では、属性ベクトル v^a と属性文脈ベクトル v^{ac} 、及びその二ベクトル間の関係のみを考慮する。このため、目的関数は、SGNS の目的関数と同様の形となる。^{*5}

属性セルの各語 w と、その水平文脈語 $z \in H_c$ それぞれに対し、スコア関数は

$$\log \sigma(v_z^{ac} \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^{ac} \cdot v_{w_k}^a)$$

と定義される。ここで、 K は Negative sample の数、 w_k

^{*5} word2vec のオリジナルの論文 [9] では、この形を、単語出現確率の最大化の式から導出しているが、本稿では、特に導出には拘らず、良いベクトル表現を得るための目的関数としてのみ扱う。

は一様分布から選ばれた疑似負例である。全体の目的関数は、これを全属性セルについて足し合わせた、以下の式となる。

$$l = \sum_{c \in C \wedge y_c=1} \sum_{z \in H_c} \left\{ \log \sigma(v_z^{ac} \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^{ac} \cdot v_{w_k}^a) \right\}$$

4.3.2 Vertical

Vertical（垂直目的関数）では、属性ベクトル v^a と、属性値ベクトル v^v 、及びその二ベクトル間の関係のみを考慮し、文脈 z は垂直文脈 V_c から選ぶ。

属性セルの各語 w と、その垂直文脈語 $z \in V_c$ それぞれに対し、スコア関数は

$$\log \sigma(v_z^v \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^v \cdot v_{w_k}^a)$$

と定義され、全体の目的関数は、これを全属性セルについて足し合わせた、以下の式となる。

$$l = \sum_{c \in C \wedge y_c=1} \sum_{z \in V_c} \left\{ \log \sigma(v_z^v \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^v \cdot v_{w_k}^a) \right\}$$

4.3.3 Cross

Cross（交差目的関数）では、水平文脈と垂直文脈を共に用いる。このため、属性ベクトル v^a 、属性文脈ベクトル v^{ac} 、属性値ベクトル v^v の三種のベクトルとその間の関係（図 1 の左上の枝に相当）を用いる。属性セルの各語 w と、その文脈語 z に対し、スコア関数は、

$$l_{(z,w)} = \begin{cases} \log \sigma(v_z^v \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^v \cdot v_{w_k}^a) & (z \text{ が垂直文脈の場合}) \\ \log \sigma(v_z^{ac} \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^{ac} \cdot v_{w_k}^a) & (z \text{ が水平文脈の場合}). \end{cases}$$

と定義され、全体の目的関数は、これらを各属性セルについて足し合わせた、以下の形となる。^{*6}

$$l = \sum_{c \in C \wedge y_c=1} \left[\sum_{z \in V_c} \left\{ \log \sigma(v_z^v \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^v \cdot v_{w_k}^a) \right\} + \sum_{z \in H_c} \left\{ \log \sigma(v_z^{ac} \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^{ac} \cdot v_{w_k}^a) \right\} \right]$$

^{*6} 属性ベクトルは、目的関数中で二種のベクトル（属性文脈ベクトル及び属性値ベクトル）と関係しているが、学習は word2vec と同様に、各ステップで 1 つの各単語・文脈ペア (w, z) のみを訓練データと見なした SGD により行っており、二項のうち一項のみが有効となるため、word2vec と同様の更新式が使えることに注意されたい。

4.3.4 Square

Square は、四種のベクトルすべてを用いるモデルであり、図1のすべての枝を考慮することに相当する。全体の目的関数は、以下で定義される。^{*7}

$$\begin{aligned}
 l = & \sum_{c \in C \wedge y_c = 1} \left[\sum_{z \in V_c} \{ \log \sigma(v_z^v \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^v \cdot v_{w_k}^a) \} \right. \\
 & + \sum_{z \in V_c} \{ \log \sigma(v_z^{vc} \cdot v_w^{ac}) + \sum_{k=1}^K \log \sigma(-v_z^{vc} \cdot v_{w_k}^{ac}) \} \\
 & + \sum_{z \in H_c} \{ \log \sigma(v_z^{ac} \cdot v_w^a) + \sum_{k=1}^K \log \sigma(-v_z^{ac} \cdot v_{w_k}^a) \} \} \\
 & + \sum_{c \in C \wedge y_c \neq 1} \sum_{z \in H_c} \{ \log \sigma(v_z^{vc} \cdot v_w^v) + \sum_{k=1}^K \log \sigma(-v_z^{vc} \cdot v_{w_k}^v) \}
 \end{aligned}$$

4.4 パラメータ学習

パラメータの学習は、word2vecと同様、一つの単語・文脈ペア (w, c) 毎に更新を行う確率的勾配降下法 (SGD) を用いる ([6])。また、やはり word2vec と同様、並列化に、データセットを分割して単純に各プロセッサを割り当て、競合を無視する Hogwild 法 ([14]) を用いた。学習の反復数は 50 とした。

5. 実験

テストデータ中、頻度 10 以上の単語についてベクトルを学習した。得られた単語ベクトルの質を評価するため、同義属性検索タスクおよび属性類推タスクを提案する。上記四種のモデルの比較のほか、比較対象として、単語をそのまま次元として用いる通常のベクトル空間モデルも用いる。また、次元数は 50、Negative Sample の数 K は 25 とした。

5.1 同義属性検索タスク

まず、単語のベクトル表現の評価として一般的に用いられている、単純な類似度の測定として、同義の属性が高い類似度を持っているか否かを測定する同義属性検索タスクを行う。2節で述べた通り、属性セルの単語は、文字列をそのまま用いている。このとき、空白の除去も行っていないため、「名前」と「名前」のように、同じ単語でも、空白の有無・その位置によって、いくつかの文字列が異なる文字列として登録されている。これらを同義語の集合と仮定し、各同義語集合 S に対し、そのうちの一つの単語 $w \in S$ をクエリとして、残りの単語を w との類似度でランキン

^{*7} 属性セルの語 w とその垂直文脈語 c に対しては、二項 (第一項と第二項) が有効になるが、二項に跨って使われているベクトルは無いため、微分は各ベクトル毎に一項のみを対象として行うことができ、word2vec と同様の更新式が使えることに注意されたい。

グする。(類似度の測定は、属性ベクトル v^a 同士の cosine 類似度を用いて行う。) S 内の他の単語 $w' \in S$ を正解と見なし、平均精度 ([4]) を用いて精度を計算する。ここで、得られた単語リストを $\langle c_1, c_2, \dots, c_n \rangle$ とし、同義語集合を $S = \{s_1, s_2, \dots\}$ とすると、平均精度とは、以下で定義される値である。

$$\frac{1}{|S|} \sum_{1 \leq k \leq n} r_k \cdot precision(k),$$

ここで、 $precision(k)$ は、上位 k 件のみを見た場合の精度 (リストに含まれる正解の割合)、 r_k は、リストの k 番目の要素が正解のとき 1、そうでないとき 0 となる値 (すなわち、 $c_k \in S$ のとき $r_k = 1$ そうでないとき $r_k = 0$) である。これは、リストを上位から見ている、正解があるたびに、その時点での精度を計算し、全正解について精度の平均をとることに相当する。定義から、最高値は 1.0 となる。これを全クエリ (いずれかの同義語集合に含まれるすべての単語) について計算し、それらの平均をとった Mean average precision (MAP) を評価指標とする。ただし、数字を含む文字列は、属性候補集合からは除いた。これは、数字を含む文字列は、単なる数字 (“5” など) や、属性の単なるバリエーション (“名前 3” など) といった、属性類似度測定において不適切と思われるものを多く含んでいたためである。この結果、クエリの総数は、2,869 個となった。^{*8}

5.2 属性類推タスク

1節で述べた通り、属性の類似度として、属性文脈を用いた類似度と共起文脈を用いた類似度が考えられる。同義属性タスクは、「属性文脈を適切に反映しているか」の測定には有効であるが、「属性の共起について適切にモデル化できているか」を測定する目的には適していないと考えられる。このため、ここでは、文章での単語埋め込みにおける単語類推タスクに類するタスクとして、属性類推タスクを新たに提案する。ここでの仮定は、「もし属性のペア (a, b) が同一の表に共起し、別の属性のペア (c, d) がまた別の表で共起しており、 $a \simeq c$ かつ $b \simeq d$ であった場合、 $b - a \simeq d - b$ が成り立ち、属性 d を、ベクトル $c + (b - a)$ と類似度の高いベクトルを探すことで発見できる」というものである。

この仮定に基づき、以下の手順でテストデータを構築する。

- (1) すべての表のうち、頻度上位 1,000 件のスキーマを取得する。ここでスキーマとは、第一行目のセルの文字列をすべて連結したものとす。
- (2) 得られたスキーマ中の属性集合を A とする。 A 中の要素のすべてのペア ($a \in A, a' \in A$) について類似度を計算し、類似度の高いペアをすべて選び出す。類似

^{*8} 数字を含む文字列を含んだ場合は、3,449 個であった。

度は、属性値の Bag of Words を用い、Vector Space Model とコサイン類似度で計算し、類似度がしきい値（ここでは 0.5）を超えたペアを類似属性ペアとする。
(3) 任意の類似属性ペアの組み合わせ $a \simeq b$ と $c \simeq d$ に関し、属性ペア (a, c) が一つのスキーマに共起し、属性ペア (b, d) がその他のスキーマに共起していた場合、クエリ $a:b - c:?$ と正解 d のペアをテスト集合に追加する。

結果として、15,248 個のクエリを得た。例えば、“氏名”：“着任” - “ゲスト”：“放送日”等の類推が得られた。

属性類推タスクの評価においても、クエリを $c+b-a$ とし、クエリとの類似度で単語を並べたリスト及び正解 d を用いて平均精度を計算する。ここでは、正解の位置を n 位としたときの、リストの 1 位から n 位までの要素での精度を計算することに相当する。これらをすべてのクエリについて平均して MAP を計算する。

5.3 結果と考察

表 2 に、両タスクの結果および、その平均と調和平均 (F-measure) を示す。“baseline” は、単語一つを一次元と見なすベクトル空間モデルによる結果を示す。ベクトルの各次元の値としては、単語頻度をそのまま用いるが、その他、pointwise mutual information (PMI) および頻度の対数 (LOG) を重み付けとして用いた場合についても同様に実験を行った。“mixed” は、水平文脈語の集合と垂直文脈語の集合を結合し、区別せずに文脈語として用い、Horizontal あるいは Vertical と同様のモデルにより学習を行った場合の結果を示す。

水平文脈のみ (Horizontal) を用いた場合は同義語タスクが、垂直文脈のみ (Vertical) を用いた場合は類推タスク

表 2 同義属性タスクおよび属性類推タスクの平均精度 (%)

手法	同義タスク	類推タスク	算術平均	F-measure
全データ				
ベースライン (ベクトル空間モデル)				
horizontal	2.46	13.04	7.75	9.72
vertical	16.29	4.74	10.52	7.34
mixed	12.96	18.04	15.5	15.08
cross	13.10	18.09	15.60	15.20
mixed (PMI)	16.08	10.28	13.18	12.54
cross (PMI)	16.61	10.23	13.42	12.66
mixed (LOG)	15.62	13.81	14.72	14.66
cross (LOG)	13.60	15.89	14.75	14.66
提案手法 (単語埋め込み)				
horizontal	1.47	12.22	6.85	2.62
vertical	20.25	3.35	11.8	5.75
mixed	14.65	15.59	15.12	15.11
cross	13.51	16.52	15.02	14.86
square	18.89	17.61	18.25	18.23

がそれぞれ低い値となり、両方のタスクで精度を得るためには、両方の文脈を用いなければならないことがわかる。また、両者のタスクの精度はトレードオフの関係にあり、例えば、“baseline” の重み付けを頻度から PMI あるいは LOG へ変化させると、類推タスクでの精度が下がる代わりに、同義語タスクでの精度が向上することが観測できる。提案手法の表形式単語埋め込みは、ベースラインに比べてやや悪い精度となった。(例えば、同じ Cross のモデル (水平文脈・垂直文脈を区別して扱う場合) の比較では、ベースラインの精度が、単純平均・調和平均ともに単語埋め込みより高くなった。) しかしながら、Square のモデルを用いることで、ベースラインの精度を単純平均・調和平均ともに上回ることが確認できた。Square は、埋め込みベクトルを用いなければ定義できないモデルのため、表形式単語埋め込みの有効性をこれにより示すことができた。

ベースライン手法は、特に類推タスクで高い精度を示した。この理由は、テストに用いた単語が頻度の高いスキーマに登場する単語であるため、類推に必要な文脈が、ベクトルによる意味表現を用いずとも、十分な数得られ、データスパースネスの問題を回避できたためであると考えられる。このため、学習のためのデータサイズを 1/10 としたデータで改めてベクトルを構築し、同様のタスクで比較を行った。^{*9}

表 3 に結果を示す。ここでは、Square の類推タスクにおける精度が若干ベースラインを上回り、また、平均・調和平均においても、大きくベースラインを上回る結果が得られた。

6. おわりに

表形式中の単語に対するベクトルを学習する、表形式単語埋め込み法を提案した。表形式特有の、水平文脈と垂直文脈を活かすためのモデルを提案し、その有効性を実験を通して確かめた。今後の課題としては、表形式単語埋め込みを、通常の記事の単語埋め込み法と統合するためのモデルの構築等が挙げられる。

謝辞 本研究は JSPS 科研費 15K00425, 15K00309,

表 3 サイズを削減したデータに対する、同義属性タスクおよび属性類推タスクの平均精度 (%)

手法	同義タスク	類推タスク	算術平均	F-measure
削減データ (1/10 Samples)				
ベースライン (ベクトル空間モデル)				
mixed	17.50	20.52	19.01	18.89
cross	17.65	20.10	18.88	18.80
提案手法 (単語埋め込み)				
cross	15.68	19.70	17.69	17.46
square	23.72	20.78	22.25	22.15

^{*9} 同義語タスクと類推タスクのテストデータ数はそれぞれ 445 および 5,124 となった。

15K16077 の助成を受けたものです。

参考文献

- [1] Bollegala, D., Alsuhaibani, M., Maehara, T. and ichi Kawarabayashi, K.: Joint Word Representation Learning Using a Corpus and a Semantic Lexicon, *Proceedings of AAAI 2016*, pp. 2690–2696 (2016).
- [2] Bollegala, D., Maehara, T., Yoshida, Y. and ichi Kawarabayashi, K.: Learning Word Representations from Relational Graphs, *Proceedings of AAAI 2015*, pp. 2146–2152 (2015).
- [3] Cafarella, M. J., Halevy, A. Y., Wang, D. Z., Wu, E. and Zhang, Y.: WebTables: exploring the power of tables on the web, *Proceedings of VLDB Endowment 1(1)*, pp. 538–549 (2008).
- [4] Chakrabarti, S.: *Mining the Web : Discovering Knowledge from Hypertext Data*, Morgan-Kaufmann Publishers (2002).
- [5] Embley, D., Hurst, M., Lopresti, D. and Nagy, G.: Table-processing paradigms: a research survey, *International Journal on Document Analysis and Recognition*, Vol. 8(2), pp. 66–86 (2006).
- [6] Ji, S., Satish, N., Li, S. and Dubey, P.: Parallelizing Word2Vec in Shared and Distributed Memory, *CoRR abs/*, Vol. 1604.04661 (2016).
- [7] Limaye, G., Sarawagi, S. and Chakrabarti, S.: Annotating and Searching Web Tables Using Entities, Types and Relationships, *Proceedings of VLDB Endowment 3(1)*, pp. 1338–1347 (2010).
- [8] Lin, Y., Liu, Z., Sun, M., Liu, Y. and Zhu, X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion, *Proceedings of AAAI 2015*, pp. 2181–2187 (2015).
- [9] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Proceedings of NIPS 2013*, pp. 3111–3119 (2013).
- [10] Munoz, E., Hogan, A. and Mileo, A.: Triplifying Wikipedia’s Tables, *Proceedings of the ISWC 2013 Workshop on Linked Data for Information Extraction* (2013).
- [11] Neelakantan, A., Roth, B. and McCallum, A.: Compositional Vector Space Models for Knowledge Base Completion, *Proceedings of ACL 2015*, pp. 156–166 (2015).
- [12] Pennington, J., Socher, R. and Manning, C. D.: Glove: Global Vectors for Word Representation, *Proceedings of EMNLP 2014*, pp. 1532–1543 (2014).
- [13] Rakesh Pimplikar, S. S.: Answering Table Queries on the Web using Column Keywords., *Proceedings of VLDB Endowment 5(10)*, pp. 908–919 (2012).
- [14] Recht, B., Re, C., Wright, S. J. and Niu, F.: Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent, *Proceedings of NIPS 2011*, pp. 693–701 (2011).
- [15] Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P. and Gamon, M.: Representing Text for Joint Embedding of Text and Knowledge Bases, *Proceedings of EMNLP 2015*, pp. 1499–1509 (2015).
- [16] Wang, Z., Zhang, J., Feng, J. and Chen, Z.: Knowledge Graph Embedding by Translating on Hyperplanes, *Proceedings of AAAI 2014*, pp. 1112–1119 (2014).
- [17] Zanibbi, R., Blostein, D. and Cordy, J. R.: A survey of table recognition, *International Journal on Document Analysis and Recognition*, Vol. 7(1), pp. 1–16 (2004).