テンソル分解に基づく述語項構造のモデル化と動詞句の表現ベクトルの学習

橋本和真　鶴岡慶雅
東京大学 工学系研究科
{hassy,tsuruoka}@logos.t.u-tokyo.ac.jp

1　はじめに

自然言語処理において, 単語の意味を表現する様々な手法が研究されており, その有効性が示されている [8]. 近年のニューラルネットワークの台頭に伴い, 単語をベクトルで表現してパラメータ化し, 大規模コーパスを用いて学習する手法が盛んに研究されている [3, 5].

このような研究では, 単語-文脈の共起の統計によって単語の表現ベクトルが計算され, 同じような文脈に出現する単語のベクトルの類似度が高くなる. また, 構文解析した大規模コーパスを利用することにより, 単語だけでなく主語-動詞-目的語といった複数単語からなる表現を学習する試みも存在する [3, 7, 9]. その中でも特に, 我々は述語項構造を利用して, 単語の表現ベクトルに加え, 複数単語の表現ベクトルを合成する関数も同時に学習する手法を研究してきた [3]. このなかで, 名詞句の学習に比べて動詞句の学習に改善の余地が大きく存在すること, また, 動詞句の表現の学習において付加部 (Adjunct) が重要な文脈情報を与える可能性が示された. 特に, 動詞句の表現の学習に関しては「述語と項の表現ベクトル間の相互作用が弱い」という部分に改善の余地がある.

そこで, 本研究では述語と項の数学的表現が互いに掛け算的に相互作用し合うモデルを考え, 動詞句の表現ベクトルの効果的な学習手法を提案する. 具体的には, 以下の二点に関する調査結果を報告する:

- テンソル分解による述語項構造のモデル化,
- 動詞句の表現の学習における付加部の有用性.

評価実験では, 提案手法の有効性を確認し, 主語-動詞-目的語の動詞句の類似度を測るタスクにおいて最高スコアを達成した.

2　テンソル分解による述語と項のモデル化

ここでは, テンソル分解を用いて述語と項の関係を掛け算的にモデル化し, 動詞句の表現を効率的に学習する手法を提案する.

述語	項 1	項 2
make	an importer	payment
in	make payment	his own domestic currency

表 1: 動詞と前置詞に関する述語項構造の例.

2.1　述語項構造に基づく共起関係のモデル化

述語項構造は, 述語とその任意個の項の関係を記述するものである. 例えば, HPSG に基づく構文解析器 Enju[1] によると, 以下の文

> An importer might be able to make payment in his own domestic currency

に関して表 1 のような述語と項の関係が得られる. Enju における述語項構造では, 動詞だけでなく任意の単語が述語として扱われる. 表 1 では, 主語と目的語に対応する名詞句を項とする他動詞 *make* に加え, 動詞句の付加部を構成する前置詞 *in* も述語として扱われている. これにより, 様々な種類の述語を介して句と句の関係が記述される.

我々は, このような述語と項の関係に基づいて, 単語の表現ベクトルと, それらを合成して複数語からなる表現のベクトルを同時に学習するモデル (PAS-CLBLM) を提案した [3]. 基本的なアイデアとして, 述語と項のうち一要素を他の要素から予測させることで学習を行った. また, 複数単語を含む項を扱う際に意味構成関数を導入し, 主語-動詞-目的語といった簡単な文の表現ベクトルを計算する関数を学習した. それにより, 動詞句の意味的類似度を測るタスクで最高スコアを達成したが, 人間が達成できる値を大きく下回っていた.

PAS-CLBLM に関する課題の一つは, 述語と項の相互作用を強めることである. PAS-CLBLM では, 述語と項が足し算的に相互作用をしており, 計算量的には軽量であり高速であるという利点がある. しかしその一方で, Tsubaki ら [7] のように, 動詞の意味が目的語によって変化する, といった性質を表現できてない.

[1] http://kmcs.nii.ac.jp/enju/.

そのような述語と項の直接的な相互作用を可能にするためには, 述語と項の数学的表現が掛け算的に強く影響し合うモデルが必要となる.

PAS-CLBLM の学習においてもう一つの重要な点は, 動詞句の表現の学習の手掛かりとして付加部を利用したことである. 例えば表 1 からは, 付加部を構成する前置詞 *in* を介して, *make payment* という行為は *his own domestic currency* によって行われる, という情報が得られる. しかし, PAS-CLBLM の学習では, 他にも形容詞-名詞などの様々な述語項構造を同時に利用していたため, このような付加部による効果が不明確であった. 付加部に基づく句と句の関係に着目することで, 動詞句 (もしくは文) の表現を学習するうえで付加部が果たし得る役割が明らかになる.

2.2 テンソル分解

まず, 述語とその 2 項の共起に関する統計情報を保持する 3 階のテンソル $\mathcal{T} \in \mathbb{R}^{|\mathbb{P}|\times|\mathbb{A}_1|\times|\mathbb{A}_2|}$ を仮定する. $\mathbb{P}$ はコーパス中の述語の集合であり, $\mathbb{A}_1, \mathbb{A}_2$ は項 1, 2 の集合である. 例えば, $\mathcal{T}$ が他動詞を扱う時, 表 1 の例では, *make* $\in \mathbb{P}$, *an importer* $\in \mathbb{A}_1$, *payment* $\in \mathbb{A}_2$ となる. また, $\mathcal{T}$ が前置詞を扱う時, 表 1 の例では, *in* $\in \mathbb{P}$, *make payment* $\in \mathbb{A}_1$, *his own domestic currency* $\in \mathbb{A}_2$ となる. このように, $\mathbb{P}$ は単語の集合であるが, $\mathbb{A}_1$ と $\mathbb{A}_2$ の要素は単語に限らない.

実際に $\mathcal{T}$ を扱うとパラメータの空間が非常に大きくなるが, Van de Cruys ら [9] の手法に従い,

$$\mathcal{T} = \mathcal{P} \times \mathbf{A}_1 \times \mathbf{A}_2 \tag{1}$$

の形で, d 次元の潜在変数を用いて $\mathcal{T}$ が暗にテンソル分解されているとする. $\mathcal{P} \in \mathbb{R}^{|\mathbb{P}|\times d\times d}$ は述語のパラメータを表す 3 階のテンソルで, $\mathbf{A}_1 \in \mathbb{R}^{d\times|\mathbb{A}_1|}, \mathbf{A}_2 \in \mathbb{R}^{d\times|\mathbb{A}_2|}$ は項 1, 2 に関するパラメータ行列である. 具体的には, $\mathcal{P}$ の各スライス $\mathbf{P}(i) \in \mathbb{R}^{d\times d}$ は各述語を表す行列であり, $\mathbf{A}_1, \mathbf{A}_2$ の各列ベクトル $\mathbf{a}_1(j), \mathbf{a}_2(k) \in \mathbb{R}^{d\times 1}$ は各項を表すベクトルである. ここで, 特定の述語項の組 (i, j, k) が与えられたとき, その組の尤もらしさ (つまり, $\mathcal{T}$ の各要素) は以下のように計算される:

$$\mathcal{T}_{i,j,k} = \mathbf{a}_1(j)^{\mathrm{T}}\mathbf{P}(i)\mathbf{a}_2(k) \tag{2}$$

Van de Cruys ら [9] は, 主語-動詞-目的語の三つ組みに関して, あらかじめ $\mathcal{T}$ を PMI により計算し, 単語ベクトルを別の手法により与えることで, 動詞のテンソル $\mathcal{P}$ を計算した. しかし, 我々の手法は,

- 式 (1) の右辺の全てを学習パラメータとしている
- 項が単語に限らない

述語	項 1	項 2
make	importer	payment
in	importer make payment	currency

表 2: 作成した学習データの例.

という点で異なる. また, このように暗に巨大なテンソルを分解するという考え方は, Levy と Goldberg [5] の, Skip-gram モデルに関する解釈と類似している [2]. 我々の手法では, 項が単語に限らないため, PAS-CLBLM のように, 意味構成関数を導入して項の表現ベクトルを計算することも可能である. つまり, 表 1 で, 前置詞 *in* の項 1 の *make payment* を表現する際には, 独立した表現ベクトルを与えるか, 2 単語 *make*, *payment* の表現から計算するか, の 2 通りが可能である.

パラメータの学習 我々のモデルのパラメータは, 尤もらしい述語項の組とそうでないものをロジスティック回帰の形で区別するように学習される. コーパスから得られた全ての述語項の組 (i, j, k) に関して, (i', j', k') を $i' \in \mathbb{P}$, $j' \in \mathbb{A}_1$, $k' \in \mathbb{A}_2$ としてサンプリングし,

$$\begin{aligned} -\log\sigma(\mathcal{T}_{i,j,k}) &- \log(1-\sigma(\mathcal{T}_{i',j,k})) \\ &- \log(1-\sigma(\mathcal{T}_{i,j',k})) \\ &- \log(1-\sigma(\mathcal{T}_{i,j,k'})) \end{aligned} \tag{3}$$

を誤差関数として定義する. σ はロジスティック関数である. この誤差関数の総和をとって目的関数として, それを最小化するように AdaGrad [1] により学習を行う. この学習により, 例えば, 同じような項をとる述語を表す行列表現が近くなることなどが期待される.

より具体的に述語を表す行列の学習過程を見ると, Grefenstette と Sadrzadeh [2] の手法との関連性が見えてくる. 彼らのモデルでは, 2 項をとる述語 (例えば, 他動詞) は, コーパス中に実際に出現した 2 項の表現ベクトルの直積の総和で表されるとした. 我々の手法でも, 式 (3) に関して, 述語の行列 $\mathbf{P}(i)$ の更新式は項のベクトルの直積を基に計算される (α は学習率):

$$\begin{aligned} \mathbf{P}(i) \leftarrow \mathbf{P}(i) - \alpha\big\{&(\sigma(\mathcal{T}_{i,j,k})-1)\mathbf{a}_1(j)\mathbf{a}_2(k)^{\mathrm{T}} + \\ &\sigma(\mathcal{T}_{i,j',k})\mathbf{a}_1(j')\mathbf{a}_2(k)^{\mathrm{T}} + \\ &\sigma(\mathcal{T}_{i,j,k'})\mathbf{a}_1(j)\mathbf{a}_2(k')^{\mathrm{T}}\big\} \end{aligned} \tag{4}$$

3 実験設定

3.1 学習データと学習の設定

British National Corpus (BNC) を学習コーパスとして用いた. まず構文解析器 Enju を用いて述語項構造

[2] $\mathcal{T}_{i,j,k}$ がどのような値に対応するのか, という点に関する説明は本稿では省略するが, 三つ組みの PMI に関連する値になることを確認した.

を取得し, 単語を基本形に直した後に表 1 のように,

(a) 2 項をとる動詞で, 項が名詞句

(b) 2 項をとる前置詞で, 項が他動詞句と名詞句

であるものを選択した. ここでは簡単のため, 項の名詞句に関しては主辞をとり, 単語にした. また, 前置詞の項 1 の他動詞句に関しては, 主語が存在する場合にはそれも含めて動詞句とした. 例えば, 表 1 からは表 2 のような学習データが作られる. BNC で, 単語の品詞を考慮した出現頻度上位 10 万単語を対象にしたところ, (a) に関しては 138 万事例 (123 万種類), (b) に関しては 93 万事例 (88 万種類) のデータが得られた.

本稿の全ての実験において, 潜在空間の次元は $d = 50$ に設定し, モデルパラメータは全て乱数で初期化した. 他のハイパーパラメータ (学習率, ミニバッチサイズなど) は, 未知データに関して式 (3) を小さくするように 5 分割交差検定によって決めた.

3.2 項の表現ベクトルの意味構成関数

本実験では, 前置詞の項 1 が主語-動詞-目的語, または動詞-目的語の動詞句になっているため, 構成要素の 2, 3 単語から項のベクトルを構成することが可能である. これに関しては様々な手法が考えられるが, ここでは特に Kartsaklis ら [4] の copy-subject という手法を用いる. copy-subject では, 動詞の行列 $\mathbf{P}(i)$ と主語, 目的語のベクトル $\mathbf{a}_1(j), \mathbf{a}_2(k)$ が与えられた時,

$$\mathbf{a}_1(j) \odot (\mathbf{P}(i)\mathbf{a}_2(k)) \tag{5}$$

として主語-動詞-目的語からなる動詞句の表現ベクトルを計算する. $\odot$ はベクトルの要素ごとの積を表す. また, $\mathbf{P}(i)\mathbf{a}_2(k)$ の部分は動詞-目的語の表現ベクトルに対応すると解釈できる. Kartsakils ら [4] は, 動詞の行列を得る手法に関して Grefenstette と Sadrzadeh [2] に基づいているため, 我々のモデルでも copy-subject を用いることが有効であると考えられる.

4 動詞句の表現ベクトルの定性的評価

4.1 意味構成関数無しの学習

ここでは, 付加部を構成する前置詞が動詞句の表現の学習に及ぼす影響を確認するため, 前置詞のみのデータを用いて学習を行った. また, 動詞句の意味構成関数は用いず, 全て単一のトークンとして扱った. 表 3 に学習結果の例を示す. 表 3 では, 全ての動詞句表現 (*run company* など) が独立にパラメータ化されて学習されており, いくつかのクエリに対してコサイン類似度が最も高いものを順に列挙した.

クエリ	類似度の高い表現
run company	operate business, carry duty, support project
win race	break record, win title, win championship
take role	become leader, play part, play role
meeting take place	hold meeting, hold race, ceremony take place

表 3: パラメータ化された (主語)-動詞-目的語の表現ベクトルの例.

クエリ	類似度の高い表現
man make payment	man pay maintenance, man pay contribution, husband make payment
director run company	director run airline, manager run firm, manager run company

表 4: copy-subject により計算された 主語-動詞-目的語の表現ベクトルの例.

表 3 を見ると, 前置詞を介した項の共起関係の情報により, 動詞句の意味的な類似度を測ることが可能な表現が学習されていることがわかる. これは, 動詞句の表現を学習するうえで, 付加部に基づく述語の項と項の関係に着目することの有効性を示している.

4.2 意味構成関数有りの学習

次に, 動詞と前置詞のデータの両方を用い, copy-subject による意味構成関数も導入して学習を行った. 表 3 と同様に, 表 4 に学習結果の例を示す. *man make payment* の例では, *make payment* が「支払いをする」という意味を持つことが表現されていることがわかる. *man make payment* と *man pay maintenance* の表現ベクトルのコサイン類似度は 0.91 であるが, *man make payment* と *man pay attention* のコサイン類似度は 0.60 であることから, 動詞の意味が目的語に依って適切に変化することが可能になっている.

5 動詞句の類似度を測るタスクによる評価

5.1 評価データ

動詞句の表現に関する定量的な評価には, Grefenstette と Sadrzadeh [2] のデータセット (GS) と Mitchell と Lapata [6] のデータセット (ML) を用いた. GS データセットでは, 2 つの他動詞が共通した主語と目的語を伴ったとき, どの程度の意味的な類似度があるか, というスコアが 200 事例に関して人手で付けられている. ML データセットでは, 動詞-目的語の組に関して意味的な類似度のスコアが 108 事例に関して人手で付けられている. 各事例には複数人がスコアを与えてい

	GS	ML
提案手法 (動)	0.417 (0.517)	0.360 (0.512)
提案手法 (動/前)	**0.451** (**0.552**)	0.434 (0.621)
PAS-CLBLM*	0.311 (0.385)	0.406 (0.613)
PAS-CLBLM [3]	0.344 (0.422)	**0.454** (**0.669**)

表 5: 各データセットにおけるスピアマンの相関係数.

る. 同一データに複数人がスコアを付けた場合の扱いに関しては, その平均値をとるという方法と, それぞれ別データ点として扱う方法の 2 通りがある. 本稿では, その両方を採用し, それらのスコアと, 各モデルが出力した動詞句の組のコサイン類似度のスピアマンの相関係数を用いて評価を行った.

5.2 比較手法

本稿における比較手法は, PAS-CLBLM である. PAS-CLBLM は, GS データセットと ML データセットで最高スコアを達成したモデルである. ここでは, 以前に報告した結果 [3] (PAS-CLBLM) に加え, 本稿で用いた学習データを使った場合の結果 (PAS-CLBLM*) も報告する. PAS-CLBLM は, 形容詞-名詞などの他の述語項構造や, bag of words も利用している.

5.3 結果

表 5 に, GS データセットと ML データセットに関するスピアマンの相関係数を示す. 表 5 中の全ての結果は, 「人手のスコアを平均しない場合 (平均する場合)」の形で示されている. また, 「提案手法 (動)」は動詞の学習データのみを用いた場合の結果であり, 「提案手法 (動/前)」は動詞と前置詞の学習データを共に用いた場合の結果である. 提案手法の動詞句の表現ベクトルは, いずれの場合も 3.2 節の方法で計算した.

まず, GS データセットに関しては, 動詞の学習データのみを用いた段階で PAS-CLBLM のスコア (0.344) を大きく上回るスコア (0.417) を達成している. これは, 表 4 で示した通り, 動詞の意味が適切に主語と目的語から影響を受ける性質を表現できているからであると考えられる. この性質は, PAS-CLBLM で用いられた意味構成関数には無かった. また, 前置詞のデータも組み合わせることで, さらにスコアが改善されたことがわかる (0.417 から 0.451).

ML データセットでは, 動詞の学習データのみでは PAS-CLBLM のスコア (0.454) を大きく下回るスコア (0.360) となった. しかし, この段階では, 式 (2) からわかる通り, 動詞-目的語の表現を学習する手がかりは「どのような主語をとりやすいか」ということだけであるから, スコアが低くても不自然ではない. そこで前置詞の学習データを追加して学習を行うと, スコアが大幅に改善された (0.360 から 0.434). これにより, 前置詞を介して項と項の関係を考慮することで, 動詞句の意味の学習に有用な情報が得られたことがわかる. ただし, 依然として PAS-CLBLM のスコア (0.454) には及ばないが, PAS-CLBLM と同様に, 形容詞-名詞などの他の述語項構造の情報の利用によってさらにスコアを改善する余地があると考えられる. 実際, 5.2 節で述べた学習データの差により, PAS-CLBLM と PAS-CLBLM*のスコアには差が出ている.

以上から, 動詞句の表現の学習において付加部が有用であることがわかる. 本稿では付加部として前置詞句のみに着目したが, 接続詞などの付加部を用いると, 動詞句と動詞句の関係 (例えば, 因果関係など) を考慮した動詞句の表現の学習が可能になると考えられる.

6 おわりに

本稿では, テンソル分解によって述語項構造をモデル化する手法を提案した. その中で, 付加部を構成する前置詞を介した項の関係に着目することで, 動詞句の表現の学習を改善し, 動詞句の意味的な類似度を測るタスクでその効果を確認した. 今後は, 述語の項が句になっていることを活かし, より長い文の表現を学習して文の言い換え認識などのタスクで評価する.

参考文献

[1] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *JMLR*, 12:2121–2159, 2011.

[2] E. Grefenstette and M. Sadrzadeh. Experimental Support for a Categorical Compositional Distributional Model of Meaning. In *EMNLP*, 2011.

[3] K. Hashimoto, P. Stenetorp, M. Miwa, and Y. Tsuruoka. Jointly Learning Word Representations and Composition Functions Using Predicate-Argument Structures. In *EMNLP*, 2014.

[4] D. Kartsaklis, M. Sadrzadeh, and S. Pulman. A Unified Sentence Space for Categorical Distributional-Compositional Semantics: Theory and Experiments. In *COLING*, 2012.

[5] O. Levy and Y. Goldberg. Neural Word Embedding as Implicit Matrix Factorization. In *NIPS*. 2014.

[6] J. Mitchell and M. Lapata. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1439, 2010.

[7] M. Tsubaki, K. Duh, M. Shimbo, and Y. Matsumoto. Modeling and Learning Semantic Co-Compositionality through Prototype Projections and Neural Networks. In *EMNLP*, 2013.

[8] J. Turian, L. Ratinov, and Y. Bengio. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *ACL*, 2010.

[9] T. Van de Cruys, T. Poibeau, and A. Korhonen. A Tensor-based Factorization Model of Semantic Compositionality. In *NAACL-HLT*, 2013.