述語項構造に基づくニューラルネットワーク言語モデルの学習

橋本 和真 † 三輪 誠 ‡ 鶴岡 慶雅 † 近山 隆 †
† 東京大学 工学系研究科, ‡ マンチェスター大学
{hassy, tsuruoka, chikayama}@logos.t.u-tokyo.ac.jp
makoto.miwa@manchester.ac.uk

1 はじめに

ニューラルネットワークを用いた言語モデル (Neural Network Language Model, NNLM) により学習された単語ベクトルは, 単語の意味的な情報を内在すると言われており, 自然言語処理の様々なタスクで有用であるとして注目されている [2, 10]. 従来の NNLM では, n グラムといった単語列をもとにしてきたが [2], 構文情報を利用して文の表現ベクトルを獲得する試みもなされている [5, 6, 10]. また, 隠れ層を伴う NNLM [2] の学習は計算コストが高く, 隠れ層を伴わずに単語ベクトルの学習を効果的に行う手法が提案されている [9]. しかし, NNLM の隠れ層が単語の意味を組み合わせた表現として有用なものになっていれば, 単語だけでなく, 句や文のベクトルも同時に学習できることになり, 様々な応用が期待できる.

本研究では, NNLM における構文情報の適用可能性と, その隠れ層の有用性を調査するため, 述語項構造 (Predicate-Argument Structure, PAS) をもとにした NNLM (PAS-NNLM) を提案する. まず PAS-NNLM により, n グラムをもとにした NNLM [2] の単語ベクトルの, 効果的な再学習が可能であることを報告する. さらに, その単語ベクトルと PAS の表現ベクトルにより, 短い句の意味的な類似度を測るタスクで高スコアを達成できることを報告する.

2 PAS-NNLM

英語では, HPSG に基づく主語-動詞-目的語, 形容詞-名詞, 名詞-名詞などの様々なカテゴリの PAS が存在する. 例えば, 'heavy rain' という句には, 述語 'heavy' とその項 'rain' から成る, 形容詞-名詞の PAS が含まれている. Tsubaki ら [10] は動詞-目的語の関係のみに着目したモデルを提案したが, 我々が提案する PAS-NNLM は, あらゆるカテゴリの PAS を利用する NNLM である. PAS-NNLM の学習により, 例えば, 名詞の単語ベクトルは, どの形容詞に修飾されやすいか, どの動詞の主語または目的語になりやすいか, などの要因によって学習が行われることが期待される.

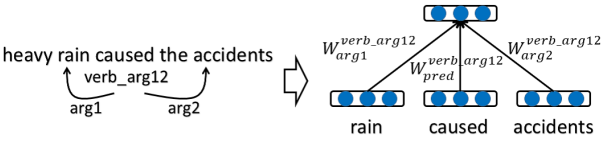

図 1: カテゴリ c =verb_arg12 (2 項をとる動詞) の述語 (pred) p =caused が項 1 (arg1) a_1 =rain と項 2 (arg2) a_2 =accidents をとる例.

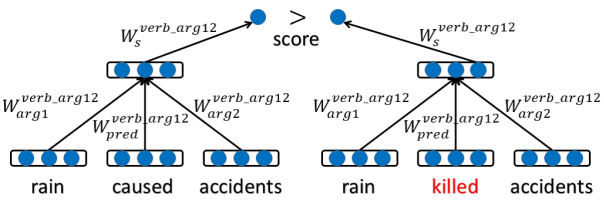

図 2: 述語 p =caused を $\tilde{p}$ =killed に置き換えた例.

2.1 PAS の表現ベクトルの計算

まず, PAS を表現するベクトルを, その述語と項の単語ベクトルから計算する. 図 1 に PAS の表現ベクトルの計算例を示す. 述語 p のカテゴリを c とし, その項を a_i とする. この PAS の表現ベクトル $v \in \mathbb{R}^{d\times 1}$ を

$$v = \tanh(W^c_{\text{pred}} V(p) + \sum_i W^c_{\text{arg}i} V(a_i)) \quad (1)$$

と計算する. ここで, $V(w) \in \mathbb{R}^{d\times 1}$ は単語 w の d 次元の表現ベクトルであり, $W^c_{\text{pred}} \in \mathbb{R}^{d\times d}$ はカテゴリ c の述語に作用する行列である. また, 同様に $W^c_{\text{arg}i} \in \mathbb{R}^{d\times d}$ はカテゴリ c の述語の項に作用する行列である. これにより, 任意のカテゴリの PAS の表現ベクトルを, 単語ベクトルと同じ次元で計算することができる.

2.2 述語の置き換えによる PAS-NNLM の学習

PAS-NNLM の学習は, Collobert らのモデル [2] (CW モデル) と同様に行う. 図 2 に例を示す. まず, 学習用のコーパスから得られる PAS の構成単語の 1 つをラ

ンダムに置き換える．ここでは, 述語を置き換え対象とし, 同じ品詞の単語からランダムに選択して置き換える．述語 p と同じ品詞の単語 $\tilde{p}$ を用いて, 式 (1) の計算を行う:

$$\tilde{v} = \tanh(W^c_{\text{pred}} V(\tilde{p}) + \sum_i W^c_{\text{arg}i} V(a_i)). \quad (2)$$

この時, 元々の PAS の表現と比較して, ランダムな置換後の PAS の表現には相対的に低いスコアを与えるスコア関数

$$score(x) = W^c_s x \quad (3)$$

を定義する．ここで, $W^c_s \in \mathbb{R}^{1\times d}$ はカテゴリ c の PAS のスコア計算用の行列である．この $score(x)$ を用いて,

$$\frac{1}{T}\sum_t \max(0, 1 - score(v_t) + score(\tilde{v_t})) \quad (4)$$

を最小化するようにモデルパラメータの最適化を行う．つまり, コーパスから得られた PAS の表現に対するスコアと, ランダムな置換により不自然になった PAS の表現に対するスコアの差が 1 以上になるようにモデルの最適化を行う．T は学習に用いる PAS の総数である．この最適化は, ニューラルネットワークの誤差逆伝播法により求めた勾配を用いて, 勾配降下法で行う．

3 PAS-NNLM の学習

3.1 学習用のコーパス: 英語版 Wikipedia

PAS-NNLM の学習用のコーパスには, 英語版 Wikipedia[1] を用いた．平文 88,122,161 文を抽出した後, PAS を利用するために構文解析器 Enju[2] で構文解析を行った．

3.2 単語-品詞ベクトル表現とその初期化

単語ベクトルは, 各単語に関して品詞タグごとに異なるベクトルを割り当てた．つまり, 名詞の train と動詞の train は別々のベクトルで表現される．我々の先行研究などでもこの手法が採られている [5, 7]．また, 単語ベクトルの次元は $d = 50$ とし, 初期化には CW モデルで学習されたもの [3] を用いた．この初期化により, PAS-NNLM の学習前は, 単語の表層が同じであれば品詞が違っても全く同じベクトルとなる．

本実験では, コーパス全体で出現頻度の上位 50,000 語を語彙として用いた [4]．ただし, 同じ単語でも品詞が違う場合には違う単語として数えた．

[1] http://dumps.wikimedia.org/enwiki/20131104/.
[2] http://kmcs.nii.ac.jp/enju/.
[3] http://ml.nec-labs.com/senna/.
[4] 実験で用いた単語リストと学習済みの単語ベクトルはダウンロード可能である: http://www.logos.t.u-tokyo.ac.jp/~hassy/publications/.

クエリ	類似度の高い単語
produce	contain, generate, create, distribute
produce_NN	spices, vegetables, beverages, cereals
produce_VB	generate, create, develop, obtain

表 1: PAS-NNLM の学習前の単語 produce, 学習後の単語 produce_NN (名詞の produce), produce_VB (動詞の produce) それぞれと類似度の高い単語の例. spices, vegetables, beverages, cereals の品詞は NNS (名詞の複数形) であり, generate, create, develop, obtain の品詞は VB である．

3.3 最適化

モデルパラメータの最適化は, 初期学習率 $\alpha = 0.1$ で AdaGrad [3] を用い, 128 事例ごとのミニバッチで行った．しかし, AdaGrad を用いたところ, 学習の初期段階でパラメータが発散することが確認されたので, 代わりに次式を用いた:

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{1 + \sum_{i=1}^t {g_i}^2}} g_t. \quad (5)$$

ここで, θ_t は t 回目の更新後のモデルパラメータであり, g_i は i 回目 $(1 \le i \le t)$ の更新時のモデルパラメータの勾配である．式 (5) により, 必ず学習率が α 以下になるため, 分母の値が小さすぎて学習率が大きくなりすぎることが無い．

本実験では, 構文解析済みの全文を用いた学習を 5 回行い, 後の実験では全て同じモデルをそのまま用いた．ただし, 語彙に含まれない単語を含む事例は使用しなかった．

4 学習後の単語ベクトルの定性的評価

PAS-NNLM の学習結果として, まずは単語ベクトルの確認を行った．表 1 に, 学習結果の例を示す．クエリとして単語ベクトルを与えた時に, コサイン距離が近いベクトルをもつ単語を表示している．特に, 品詞の情報の効果を確認するため, 単語の表層が同じで, 動詞にも名詞にもなり得る単語に関して確認した．その結果, CW モデルの単語ベクトルで全く同じに初期化された状態から, 動詞的な使い方, 名詞的な使い方, それぞれの影響を受けて学習されたことが確認できた．また, CW モデルの単語ベクトルで初期化した状態では上位に無かった単語が現れたことから, n グラムとは違う観点から学習されたと考えられる．しかし, 依然として語義の曖昧性に関する問題は残っており, それは今後の課題とする．

5 短い句の意味的な類似度を測るタスクによる評価

5.1 データセット

本実験では, 4 つのデータセットを用いた. それぞれのデータセットで, 人手で付けられた短い句の意味的な類似度のスコアと, システムが出力した類似度のスコアに関して, スピアマンの順位相関係数を計算することにより評価を行った.

主語-動詞-目的語 Grefenstette と Sadrzadeh [4] のデータセットは, 主語-動詞-目的語 (Subject-Verb-Object, SVO) の組み合わせと動詞のペアに関して, その意味的な類似度が人手でスコア付け (1-7) されているものである [5]. 例えば, [student, write, name] という SVO と動詞 spell の類似度は 7 (高い) で, [student, write, paper] と spell の類似度は 2 (低い) である. データ数は 199 事例である.

形容詞-名詞, 名詞-名詞, 動詞-目的語 Mitchell と Lapata [8] の 3 つのデータセット [6] は, 形容詞-名詞 (Adjective-Noun, AN), 名詞-名詞 (Noun-Noun, NN), 動詞-目的語 (Verb-Object, VO) それぞれの組み合わせのペアに関して, その意味的な類似度が人手でスコア付け (1-7) されているものである. データ数は, それぞれ 108 事例である.

5.2 句の表現のモデル

各データセットの短い句を表現するたに, 本実験で用いたモデルを説明する. 句, 単語の表現ベクトルの類似度の指標としては, コサイン距離を採用した.

単語ベクトルの足し算 2, 3 語程度から成る句の表現ベクトルの計算法としてよく用いられているのが, 構成単語のベクトル表現の和をとることである. Tsubaki ら [10] は, SVO の表現の計算に関してこの手法を用いた. CW モデルの単語ベクトルで初期化した状態における単語ベクトルの足し算を, 'CW add' と呼ぶ. また, PAS-NNLM の学習後の単語ベクトルの足し算を, 'PAS-NNLM add' と呼ぶ.

学習したニューラルネットワークによる表現 式 (1) を用いて PAS の表現ベクトルを計算することができる. これを 'PAS-NNLM comp' と呼ぶ. 例えば [student, write, name] の SVO を表現する際には, 式 (1) で, $c = \text{verb\_arg12}, p = \text{write}, a_1 = \text{student}, a_2 = \text{name}$ とする. また, 同様に AN, NN に関しては $c = \text{adj\_arg1}$, $c = \text{noun\_arg1}$ を用いた. VO に関しては, SVO の場合と同様の $c = \text{verb\_arg12}$ を用い, arg1 (主語) の入力を 0 にした. これらは Enju で用いられている PAS のカテゴリである. ただし入力の単語ベクトルに関しては, 実際には, 名詞は品詞 NN が付いているもの, 形容詞は品詞 JJ が付いているもの, 動詞は品詞 VB が付いているものをそれぞれ用いた.

[5] http://www.cs.ox.ac.uk/activities/compdistmeaning/GS2011data.txt.

[6] http://homepages.inf.ed.ac.uk/s0453356/share.

手法	SVO	AN	NN	VO
Human	0.62	0.52	0.49	0.55
CW add	0.14	0.46	0.55	0.46
PAS-NNLM add	0.26	**0.52**	**0.60**	0.55
PAS-NNLM comp	0.42	0.43	0.37	**0.58**
C-NLM [10]	0.38	NA	NA	NA
CoC-NLM [10]	**0.47**	NA	NA	NA
CCAE [6]	NA	0.41	0.44	0.34
SDS [1]	NA	0.48	0.50	0.35
Tensor (WSD) [7]	NA	NA	NA	0.45

表 2: 各データセットのスピアマン相関係数の比較.

5.3 結果

表 2 に, 4 つのデータセットに関するスピアマンの相関係数を示す. 本実験での結果に加え, 比較対象としていくつかの先行研究の実験結果も引用した. 特に, Hermann と Blunsom [6] の CCAE は, CCG パーザから得られる構文情報を用いた Deep Learning のモデルであり, 用いている構文情報は本研究と類似している. また, 'Human' は, 各データセットにおける複数人によるアノテーションの相関係数である.

単語ベクトルの質の向上 'CW add' と 'PAS-NNLM add' の結果を比較すると, 4 つのデータセット全てに関して相関係数が向上していることがわかる. CW モデルが n グラムをもとに, Wikipedia を用いて長い時間をかけて学習したものであることを考えると, PAS-NNLM により n グラムとは異なる観点から効果的な再学習が行えたと考えられる.

AN, NN, VO の 3 つのデータセットに関しては, 'PAS-NNLM add' により最高水準の相関係数 (0.52, 0.60, 0.55) を得た. さらに, 各データセットの 'Human' 以上の相関係数を達成した. この値を超えた後には, 相関係数の大小に関して論じるのは意味が無いと言える.

また, 単語ベクトルに品詞タグの情報を組み込んでいることも好結果の 1 つの要因と考えられる. 実際, Kartsaklis と Sadrzadeh [7] の 'Tensor (WSD)' でも,

テンソルを用いた複雑なモデルに加え, 単語の語義曖昧性解消 (Word Sense Disambiguation, WSD) の前処理として品詞の情報を用いている. 'Tensor (WSD)' は VO のデータセットに関して最高水準の相関係数 (0.45) を得たが, 本実験の 'PAS-NNLM add' は更に良い結果 (0.55) を達成した.

PAS-NNLM のモデルとしての表現力　SVO に関しては, 本実験では PAS-NNLM の学習後のモデルを用いた 'PAS-NNLM comp' により, 'PAS-NNLM add' よりも高い相関係数 (0.42) を得た. これは Tsubaki ら [10] の C-NLM による結果 (0.38) を上回るものである. この C-NLM は, 動詞と目的語のペアのみに着目して NNLM を学習するモデルである. しかし, 同じく Tsubaki ら [10] の CoC-NLM による相関係数 (0.47) には及ばなかった. CoC-NLM の優れている点は, 一種の語義曖昧性解消の手法を, 動詞-目的語の組み合わせの意味構成に取り入れたことである. また, Kartsaklis と Sadrzadeh [7] も, 単語からの意味構成の前に語義曖昧性の解消を行うことの重要性を示している. 本研究の PAS-NNLM では, 品詞の情報は用いているものの, その他の明示的な語義曖昧性の解消を行っていない. より文脈に依存した語義曖昧性解消の手法を取り入れることにより, さらなる性能向上が考えられる.

VO においても, 'PAS-NNLM comp' の相関係数 (0.58) が 'PAS-NNLM add' の結果 (0.55) を上回り, このデータセットの 'Human' (0.55) を超えた. ただし, 'PAS-NNLM add' の時点で 'Human' と同等なため, それよりも悪くはない, という程度のことが言える. SVO の場合と合わせて考えると, 動詞が関わる場合には, 'PAS-NNLM comp' は 'PAS-NNLM add' よりも良い表現力を持ち得ると考えられる.

その一方で, AN と NN に関しては, 'PAS-NNLM comp' による相関係数が, 'PAS-NNLM add' のものに劣っていた. 全く意味の無い表現になっているわけではなく, AN に関する相関係数 (0.43) は CCAE による結果 (0.41) を上回っている. しかし, 本研究の PAS-NNLM の学習では, 学習中に PAS の表現ベクトルの層が受ける影響は, あくまで言語モデルのスコアに関するものだけであるので, 必ずしも PAS の表現が高い性能を発揮するとは限らない.

現時点では, 単語ベクトルの足し算による結果が非常に良いが, より長い文を扱う際には, 足し算では表現力が不足することが予想される.

6　おわりに

本稿では, 述語項構造に着目した PAS-NNLM の学習法の提案と, その結果について報告した. まず, 構文情報を取り入れることで, 従来のモデルの単語ベクトルの効果的な再学習の可能性を示した. さらに, PAS-NNLM の隠れ層として表現されているベクトルが, 短い句の意味的な類似度を測るタスクにおいて有用になり得ることも示した. この結果は, 述語項構造などの構文情報を用いて, より長い文の意味を表現するモデルに拡張する際の基礎になることが期待される.

参考文献

[1] W. Blacoe and M. Lapata. 2012. *A Comparison of Vector-based Representations for Semantic Composition.* In *EMNLP/CoNLL.*

[2] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. *Natural Language Processing (almost) from Scratch.* In *JMLR.*

[3] J. Duchi, E. Hazan, and Y. Singer. 2011. *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.* In *JMLR.*

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

[5] K. Hashimoto, M. Miwa, Y. Tsuruoka, and T. Chikayama. 2013. *Simple Customization of Recursive Neural Networks for Semantic Relation Classification.* In *EMNLP.*

[6] K. M. Hermann and P. Blunsom. 2013. *The Role of Syntax in Vector Space Models of Compositional Semantics.* In *ACL.*

[7] D. Kartsaklis and M. Sadrzadeh. 2013. *Prior Disambiguation of Word Tensors for Constructing Sentence Vectors.* In *EMNLP.*

[8] J. Mitchell and M. Lapata. 2010. *Composition in distributional models of semantics.* In *Cognitive Science.*

[9] A. Mnih and K. Kavukcuoglu. 2013. *Learning word embeddings efficiently with noise-contrastive estimation.* In *NIPS.*

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