

潜在変数モデルを用いた同義・反義関係識別

城内 聡志 菊井 玄一郎

岡山県立大学

{cd30021r, kikui}@cse.oka-pu.ac.jp

1 はじめに

自然言語処理では、単語の意味を表すために共起分布情報から作られた単語ベクトルが直接、あるいは Neural Network の入力として頻繁に用いられている。しかし、共起分布情報に基づく単語ベクトルは単語間の「類似性」は捉えられるものの、上位下位関係、反義関係、所属関係のようなより詳細な関係の同定は難しいことが指摘されている [5]。これらのうち単語ペアが同義関係にあるか、反義関係にあるかを識別すること (以下、「同義・反義関係識別」とする) は反義関係にある単語ペアの文脈が類似していることから特に難しい。同義・反義関係識別する上で、識別対象の単語ペアの各文脈共起語のうち、一方にしか出現しない単語の情報が有効であることが知られている [8]、そこで本研究では、このような単語の情報を Neural Network を使った同義・反義関係識別への入力 (ベクトル) に変える潜在変数モデルを提案する。さらに、反義関係の反推移性 (antitransitivity)¹ を考慮して同義・反義関係識別を行う既存の Neural Network モデルについて、事前学習を不要とする改善方法を示す。

2 既存手法

Nguyen ら [6] は、反義関係の単語ペアは同義関係の単語ペアよりも同じ文で頻繁に出現するという仮説から、識別対象の単語ペアを含む文に出現するパターンを LSTM に入力し、特徴に使う AntSynNET を提案した。このモデルは、名詞の同義・反義関係識別では高性能である。しかし、各単語ペアでパターンを集めるなければならないことが問題となる。

Etcheverry ら [3] は、反推移性を捉えるために、Siamese Network を改良した PN (Prasiamese Network) を提案した。このモデルは Nguyen らが使用していたパターンや品詞の情報は使わず、単語ベクトルのみを入力で AntSynNET と同等、もしくはそれ以上の性能をもつ。しかし、このモデルはまず事前学習として同義関係識別をしなければ性能が出ないという問題がある。

¹反義関係を A 、同義関係を S とすると、反義語の反義語は同義語になるという反推移性は $\forall a, b, c : aAb \wedge bAc \rightarrow aSc$ と表せる。

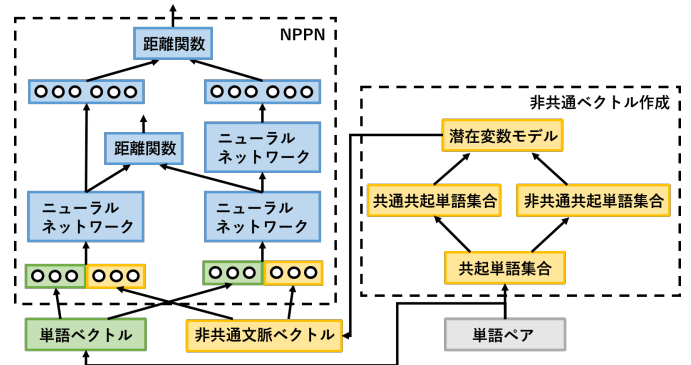


図 1: 提案手法の全体

3 提案手法

提案手法全体は、図 1 に示すように同義・反義関係識別に有効な特徴ベクトルである非共通ベクトルを作成する部分と、ベクトルを入力して同義・反義関係識別を行う NPPN (Non Pretrained Parasiamese Network)² の二つからなる。前者の中心は潜在変数モデルであり、識別する単語ペアの各単語に対してこれらの文脈の違いを捉えたベクトルを一つずつ作成する。NPPN は、既存の最高性能のモデル、PN を改良した同義・反義関係識別モデルである。以下、これらを順に説明する。

3.1 潜在変数モデル

反義関係にある単語ペアの各単語の出現する文脈は一部以外は類似している。反義関係を識別するには、この「類似していない一部の文脈」一部の文脈が有効であると考えられる。以下では単語ペアを構成する各単語の文脈のうち、他方の文脈と共通しない部分を「非共通文脈」と呼ぶ。我々は非共通文脈が単語ペアの他方の単語との意味の差に関わる部分とそれ以外の (偶然) 文脈に出現した部分から構成されると考え、Camacho-Collados ら [2] の潜在変数モデルと類似したモデルにより、前者の情報をベクトル化した「非共通文脈ベクトル」を推定する本章では、まず既存研究の、談話ベクトルを求めるための潜在変数モデルについて述べ、次に提案する同義・反義関係識別に使う非共通ベクトルを求めるための潜在変数モデルについて述べる。

²図 1 中の NPPN の Neural Network は同一である。

3.1.1 関係ベクトル

Camacho-Collados ら [2] は Arora ら [1] の文脈モデルから、関係ベクトルという二つの単語の関係を表現するベクトルをモデル化した。

Arora ら [1] の文脈モデルによると文 $S = \{w_1, \dots, w_k\}$ の単語 w_i の出現確率は、

$$P(w_i|c_S) = \alpha P(w_i) + (1 - \alpha) \frac{1}{Z_s} \exp(v_{w_i} \cdot c_S) \quad (1)$$

となる。ここで、 c_i はノルムが 1 となる談話ベクトル、 v_{w_i} は単語 w_i の単語ベクトル、 $Z_s = \sum_{w \in V} \exp(v_w \cdot c_S)$ は正規化定数、 α はハイパーパラメータである。これは、次の生成モデル等価である。

$$P(Z_i = 1) = \alpha \quad (2)$$

$$P(W_i = w|Z_i = 1) = P(w) \quad (3)$$

$$P(W_i = w|Z_i = 0) = \exp(v_{w_i} \cdot c_S) \quad (4)$$

ここで Z_i は i 番目の単語が文の文脈に依存するか、当該言語の一般的な単語分布に依存するかを示す潜在変数である。この定式化から、Camacho-Collados ら [2] は単語 a, b が共起する文 S 中に出現する単語 w_i の潜在変数 Z の値として $Z = \{z_a, z_b, z_{ab}, z_*\}$ の 4 つを考え、 z_{ab} に対応する文脈ベクトル c_{ab} を a と b の関係を示すベクトル (関係ベクトル) であるとみなしている。 c_{ab} を求めるためには z_{ab} に関する確率分布が必要なため彼らは EM 法を用いた c_{ab} の推定手法を提案している。

3.1.2 非共通文脈ベクトル

反義関係にある単語ペアは同義関係にある単語ペアと比べて共起しやすいことから、関係ベクトルは単語ペアの反義関係を表現しうが、単語ペアごとに共起単語を集める必要があることや、単語ペアがある程度の頻度で共起するコーパスが必要であるという問題がある。本研究ではこの問題を解決するために、非共通文脈ベクトルを提案する。非共通文脈ベクトルは与えられた単語ペア a, b それぞれに対して作成され、お互いを区別する特徴を表すベクトルである。非共通文脈ベクトルの作成方法は、関係ベクトルと類似している。違いは、作成時には、 a, b それぞれと共起する単語のリスト S_a, S_b のみを必要とし、 a, b で共起している文脈のデータは不要である点と、潜在変数 Z の値は $Z = \{z_{ab}, z_{a\bar{b}}, z_*\}$ の 3 つを考える点である。以下では、単語 a に対する非共通文脈ベクトル $c_{a\bar{b}}$ の作成方法について述べる。単語 b については a と b を入れ替えれば同様に求められる。

まず、 S_b には入っておらず S_a には入っている共起単語のリストを $S_{a\bar{b}} = \{w_1, w_2, \dots, w_k\}$ とする。同一単語が m 回文脈に出現する場合、 $S_{a\bar{b}}$ にはその単語が m 個含まれる。

$S_{a\bar{b}}$ から b に関連する単語³、偶然出現した単語が

³ $S_{a\bar{b}}$ には S_b 内の単語を除いたが、 b に関連するもの全てを除くことができるとは限らない。

全て除かれているとは限らない。そこで本研究では文脈語 $w_i \in S_{a\bar{b}}$ を、1) a, b 両方に関連するもの、2) a のみに関連するもの、3) 偶然文脈に出現したものの、いずれかであると考えて、次の生成モデルを考える。

$$P(W = w|Z = z)P(Z = z) \quad (5)$$

$$P(W = w|Z = z_x) \propto \exp(v_w \cdot c_x) \quad (6)$$

ここで、 z_x の値は $z_{ab}, z_{a\bar{b}}, z_*$ のいずれかであり、それぞれ、 a と b に関連するトピック、 a には関連するが、 b には関連しないトピック、 a とは直接関連しないトピックを表す。同様に、 c_{ab} は a, b に共通する文脈を特徴づけるベクトル、 $c_{a\bar{b}}$ は a のみの文脈を特徴づけるベクトル、 c_* は a には直接関係しないベクトルである。3 つのベクトルのうち、 $c_{a\bar{b}}$ が b にはない a の特徴を表すベクトルであるため、これを同義・反義関係識別用の単語 a の特徴として利用する。

以上の定式化において 3 つのベクトル $c_{a\bar{b}}, c_{ab}, c_*$ が出現する。これらのうち、 c_{ab} は共起単語集合 S_a, S_b 内の共通する単語のベクトルの平均、 c_* はコーパス中の全単語のベクトルの平均に固定し、 $c_{a\bar{b}}$ は EM 法で求める。EM 法による $c_{a\bar{b}}$ の求め方は Camacho-Collados ら [2] と同様であるため省略する。 $c_{a\bar{b}}$ は EM 法によって推定された $P(Z_i = z_{a\bar{b}}|W_i = w_i)$ を用いて以下の式で求める。

$$c_{a\bar{b}} = \text{norm} \left(\sum_{i=1}^k v_{w_i} \cdot P(Z_i = z_{a\bar{b}}|W_i = w_i) \right) \quad (7)$$

最後に、求めた非共通ベクトル $c_{a\bar{b}}, c_{\bar{a}b}$ を、単語 a, b の各単語ベクトルに接続して、次に述べる反義関係識別モデルの入力にする。

3.2 Non Pre-trained Parasiamese Network

PN は同義関係か反義関係かを識別したい単語ペアのうち一方の単語は一度だけ Neural Network に通し、もう一方は二度 Neural Network に通す。これにより、入力する際は反義関係だった単語ペアは、出力時には同義関係となる。そして、入力した単語ペア a, b が反義関係の場合は出力した表現が近づくように、同義関係の場合は遠ざかるように学習する。Neural Network を $F: R^n \rightarrow R^n$ 、距離関数を d とし、ポジティブペア集合 P に反義関係の単語ペアを入れ、ネガティブペア集合 N に同義関係の単語ペアが入れたとき、PN の損失関数は以下の式になる。

$$L_{\text{ant}} = \sum_{(a,b) \in P} \max\{d(F(a), F(F(b))) - \beta, 0\} + \sum_{(a',b') \in N} \max\{0, \alpha - d(F(a'), F(F(b')))\} \quad (8)$$

α はネガティブペアの閾値、 β はポジティブペアの閾値を表すパラメータ。PN は、この損失関数で学習する

前に、識別対象の単語ペアの各単語を一回だけ Neural Network に通す Siamese Network で同義語識別の学習をした Neural Network を使用する必要がある。

本研究で提案する NPPN(Non Pre-trained Parasiamese Network) は事前学習をなくすために、モデルの中間表現に制限をかける項を損失関数に追加した。つまり、単語ペアの各単語を一回だけ Neural Network に通した段階で、単語ペアが同義関係の場合は出力された表現を近づけ、反義関係の場合は遠ざけるように学習することで同義語識別も同時に行う。追加する項 L_{mid} は以下の式になる。

$$L_{mid} = \sum_{(a,b) \in P} \max\{0, \alpha - d(F(a), F(b))\} + \sum_{(a',b') \in N} \max\{d(F(a'), F(b')) - \beta, 0\} \quad (9)$$

この項を追加することで、一度の学習で反義語の反推移性を捉え、性能を向上させることが期待できる。

4 実験

提案する識別モデルの性能を調べるために、本研究では反義関係、同義関係の単語ペアを識別するタスクを用いて評価する。

4.1 同義・反義関係識別データセット

本研究は、Nguyen ら [7] の作ったデータセットを使用した。このデータは、形容詞、動詞、名詞の3つの品詞で分かれており、WordNet と WordNik から抽出された単語ペアからできている。同義・反義関係ペアの比率は同じで、品詞ごとに学習、開発、テストデータに分けられている。詳細は表1に示す。また、このデータには単語ペアの順番を入れ替えたものが入っていないため、入れ替えた単語ペアを加える反転操作を学習データに行い、開発、テストデータはそのまま使用した。評価は Nguyen ら [7] と同様に、Precision, Recall, F1 値で行う。

	学習	開発	テスト
形容詞	5562	398	1986
動詞	2534	182	908
名詞	2836	206	1020

表 1: 同義・反義関係識別タスクのデータの詳細

4.2 事前学習ベクトル

非共通ベクトルの作成と、NPPN の入力に用いる単語ベクトルは、English Wikipedia[4] から fasttext によって学習したものを使用した。共起単語の作成にも English Wikipedia[4] を使用した。EM 法のステップ

数は 3、共起単語は window size 10 に出現する単語のうち、頻度数の上位 150 以上のもののみを使用した。

4.3 比較モデル

ベースラインとして使用したのは FNN(Feed-forward Neural Network) 分類器で、ベクトルを、各次元が [600, 400, 200, 1] の4つの層に入力し、反義語か同義語かを [0, 1] で出力する。最後の層は出力時に sigmoid 関数を通し、それ以外は出力時に ReLU 関数に通す。学習には二値交差エントロピーと Adam を使用した。

さらに、二つのモデルを比較対象として使用した。一つ目は Nguyen ら [6] が提案した AntSynNet である。これは、単語ベクトルに加えて、LSTM を使って単語間のパターンを入力する。さらに、品詞、形態情報等の情報も加える。二つ目は、Etchevrry ら [3] の提案した PN である。

4.4 パラメータ

モデルのパラメータは以下のものから選んだ。

- ・学習率 {0.1, 0.007, 0.005, 0.003, 0.001, 0.0007, 0.0005, 0.0003, 0.0001}
- ・ポジティブペアの閾値 {0.5, 0.7, 0.9}
- ・ネガティブペアの閾値 {0, 0.3, 0.5}
- ・バッチサイズ {64, 128, 256, 384, 512}

ネットワークの weight と bias は 0 で初期化し、NPPN の距離関数には cos 類似度を使用した。NPPN と PN の Neural Network の次元数は入力するベクトルの次元は、[600, 360, 180, 600] にし、最後の層は出力時に tanh 関数を通し、それ以外は出力時に ReLU 関数に通す。上記のパラメータ選択には、Grid Search を使用し、開発データで F1 値が最も高くなるものを各モデル、各品詞ごとに選択した。実装は Pytorch で行った。

4.5 結果と考察

実験した結果を表2に示す。表2の結果から、名詞では、ベースラインに非共通ベクトルを加えたモデルが既存の最高性能モデルと同じ性能を持っていることがわかる。また、NPPN の場合も非共通ベクトルを加えることで精度が向上している。これは、Santus ら [8] が述べている、共起単語の違いによる影響をもっとも受ける品詞は名詞という考察と一致している。

NPPN はどの品詞タイプでも PN と同等、もしくはそれ以上の性能を達成しており、追加した中間の損失項が識別性能を向上させていることが分かる。

なお、既存研究の FNN の性能は他の手法と比較すると低いが、学習データの反転操作により、他のモデルと同等の性能がでていることが確認できた。これより、反転操作は既存研究の PN と同等の性能をもたらす操作だと考えられる。

Model	形容詞			動詞			名詞		
	P	R	F1	P	R	F1	P	R	F1
FNN	0.691	0.664	0.667	0.756	0.641	0.694	0.764	0.716	0.739
AntSynNET	0.763	0.807	0.784	0.743	0.815	0.777	0.816	0.898	0.855
PN	0.855	0.857	0.856	0.864	0.921	0.891	0.837	0.859	0.848
FNN*(Word)	0.848	0.840	0.844	0.884	0.881	0.882	0.853	0.832	0.841
FNN*(Word+Context)	0.850	0.839	0.845	0.884	0.885	0.885	0.854	0.856	0.855
NPPN*(Word)	0.855	0.876	0.864	0.898	0.869	0.883	0.829	0.768	0.796
NPPN*(Word+Context)	0.867	0.887	0.876	0.896	0.900	0.898	0.858	0.766	0.808

表 2: 評価データにおける, 同義・反義関係識別の Precision, Recall, F1 値. *は実装したモデル. Word は入力単語ベクトルのみ, Word+Context は入力単語ベクトルと非共通文脈ベクトルを concatenation したベクトル.

	ベクトル	共起単語
hot	単語	bubbling, billboard, cool, cold, chart, warm, chilly, cooler
	非共通文脈	peak, billboard, chart, oricon, tracks, peaking, top, bubbling
cool	単語	cooler, warm, cooling, cold, temperture, dry, chill, hot
	非共通文脈	exhaut, idle, saturate, airflow, timing, fit, inject, stationary

表 3: 単語ベクトルと非共通文脈ベクトルの最近傍単語.

4.6 ベクトルの分析

反義関係の単語ペアの潜在変数モデルによる非共通ベクトルの各々が反対の意味を排除できているかを調べるために周辺にあるベクトルを見る. 表 3 に「hot」と「cold」の単語ベクトルの最近傍にある単語と非共通ベクトルの最近傍にある単語を示す. 表 3 より, 「hot」の場合, 「peak」, 「billboard」, 「bubbling」, 「top」のような熱さを思い浮かばせる単語が周りに来ており, 単語ベクトルの場合と違い, 寒さを感じさせるものは見られない. 「cool」の場合は, 「exhaut」, 「idle」のような動かないことや, 「saturate」のような冷たさを感じさせる単語が周りに来ており, 熱さを思い浮かばせる単語はあまり見られなかった. これより, 本研究で提案した非共通ベクトルは反対の意味を排除できていることが定性的に分かった.

5 まとめ

本稿では, 反義語識別をする際に有効な特徴を作るための潜在変数モデルと事前学習を必要としない NPPN について提案した. 一つ目に提案した潜在変数モデルは, 各単語の共起単語を使って単語ペアのそれぞれに依存していない意味を見つけるためのフレームワークである. このモデルの利点は, パターンや関

係から作られた特徴を用いないため, 単語間ごとに共起単語を求める必要がなく, 名詞の反義語を識別する際に有用な特徴である点である. 二つ目に提案した NPPN の良い点は事前学習を必要とせず, PN と同等, もしくはそれ以上の性能をもっている点である. また, 反義語識別における学習データへの反転操作の効果も実験から確認できた.

謝辞

本研究は JSPS 科研費 16K00441 の助成を受けた.

参考文献

- [1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*, 2017.
- [2] Jose Camacho-Collados, Luis Espinosa-Anke, Shoaib Jameel, and Steven Schockaert. A latent variable model for learning distributional relation vectors. In *IJCAI*, pp. 4911–4917, 2019.
- [3] Mathias Etcheverry, Dina Wonsever. Unraveling antonym’s word vectors through a siamese-like network. In *ACL*, pp. 3297–3307, 2019.
- [4] Mandar Joshi, Eunsol Choi, Omer Levy, Daniel Weld, and Luke Zettlemoyer. pair2vec: Compositional word-pair embeddings for cross-sentence inference. In *NAACL*, pp. 3597–3608, 2019.
- [5] Saif Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. Computing lexical constraint. *CL*, Vol. 39, pp. 555–590, 2013.
- [6] Kim Anh Nguye, Sabine Schulte im Wald, and Ngoc Thang Vu. Distinguishing antonyms and synonyms in a pattern-based neural network. In *EACL*, pp. 76–85, 2017.
- [7] Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. Integrating distributional lexical contrast into word embeddings for antonym synonym distinction. In *ACL*, pp. 454–459, 2016.
- [8] Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang. Taking antonymy mask off in vector space. In *PACLIC*, pp. 135–134, 2014.