

単語埋め込みにおける複数視点の対義語判定

Distinguish Antonyms in Multiple Viewpoints on Distributed Word Representations

中村 拓^{*1} 田然^{*2} 佐々木 翔大^{*1} 乾 健太郎^{*1*3}
Taku Nakamura Ran Tian Shota Sasaki Kentaro Inui

^{*1}東北大学 ^{*2}産業技術総合研究所
Tohoku University National Institute of Advanced Industrial Science and Technology (AIST)

^{*3}理化学研究所 AIP センター
RIKEN Center for Advanced Intelligence Project (AIP)

Distributed representations of words (word embeddings) are proven effective for measuring word similarity and additive compositionality. However, it is challenging to distinguish between synonyms and antonyms because their proximity in the word embedding space is generally close due to their interchangeability in the context. In this paper, we aim to build a model that can discriminate antonyms in the embedding space without affecting important properties of original word embeddings. Our method is designed to learn multiple hyperplanes for capturing antonymy relations in the embedding space in a supervised manner. Our empirical evaluation demonstrates that we can reasonably distinguish between synonyms and antonyms and reveals that several intriguing issues still remain.

1. はじめに

単語の意味をベクトル空間に埋め込む単語埋め込みは、自然言語処理の幅広いタスクに応用されている [Chen 14, Melamud 16]. 単語の周辺文脈を予測することで学習された単語埋め込みでは、似た文脈で出現する単語同士がベクトル空間内において近い位置に埋め込まれるため (図 1 中の “happy” と “glad”), 単語の類似性判定に有効であることが示されている [Hill 15]. しかしながら、対義語のように異なる極性を持つにも関わらず似た文脈で出現する単語については、従来の埋め込み手法を用いると、やはりベクトル空間内で近い位置に埋め込まれてしまい、単語間の同義・類義関係と対義関係を区別することができないという問題がある. 例えば、感情極性を持つ “happy” と “unhappy” は、どちらの単語も人の感情を言及する文脈で出現することが多く、ベクトル空間内で近い位置に埋め込まれてしまう (図 1). この問題は単語埋め込みを含意関係認識などの自然言語の意味計算に適用する上で大きな障害となってきた [Nguyen 17].

単語埋め込みにおける対義語の扱いについてはこれまで大きく 2 つのアプローチが提案されている. 1 つ目は辞書情報を教師情報として利用し、ベクトル空間内において同じ極性の単語を近づけ、異なる極性の単語を遠ざけるように単語埋め込みを学習するアプローチである [Faruqui 15, Ono 15, Pham 15, Mrksić 16]. しかし、対義語のベクトルを大きく遠ざけると、元の単語埋め込みベクトルが持っていた加法構成性などの重要な性質を損なうおそれがあり、実際にそうした副作用が報告されている.

第 2 のアプローチは、元の単語埋め込みのベクトルは変えずに、感情極性などの極性を識別する識別面を学習する手法である. この枠組みでは、元の単語埋め込みが持つ加法構成性などの重要な性質を損なうことなく “happy”-“unhappy” の

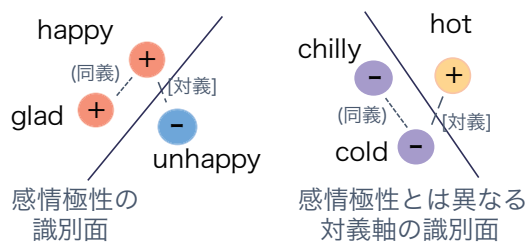


図 1: 単語埋め込みベクトル空間における複数の識別面 (対義識別面) の学習

ような対義関係を表現できる. Rothe ら [Rothe 16] は単語埋め込み空間の直交部分空間に感情極性など特定の意味的な性質を写像し、他の意味的な特徴から分離できることを示した. また、中村ら [Nakamura 18] は、単語埋め込み空間内で個々単語の感情極性を識別する線形識別器を学習することができることを実験的に示している. 中村らの方法では、例えば図 1 の “happy”, “unhappy”, “glad” のようにベクトル空間で近傍にある語の感情極性を識別する線形識別器を、一部の単語に対する感情極性ラベル付き訓練データで学習することで、感情極性が未知の単語について 90 % を超える精度で感情極性を予測している. 極性の識別面は線形識別平面であるため、この平面の法線ベクトルが埋め込み空間のどれかの基底と一致するように空間全体を回転させれば、特定の 1 次元の成分の符号を見るだけで極性を判別できる単語埋め込み空間が得られることになる.

このように中村ら [Nakamura 18] のアプローチにはいくつかの重要な利点があるが、一方でこれを感情極性以外の一般の対義関係に一般化するには、次に述べる課題を解決する必要がある. 中村らの手法では、単語埋め込み空間を感情極性がポジティブの領域とネガティブの領域に単一の線形識別平面で分離することができるという仮定をおいていた. しかしながら、一般の対義関係の中には、例えば図 1 の “hot”-“cold” の対義ベ

連絡先:

中村 拓 : tnakamura@ecei.tohoku.ac.jp
田 然 : robin.tianran@gmail.com
佐々木 翔大 : sasaki.shota@ecei.tohoku.ac.jp
乾 健太郎 : inui@ecei.tohoku.ac.jp

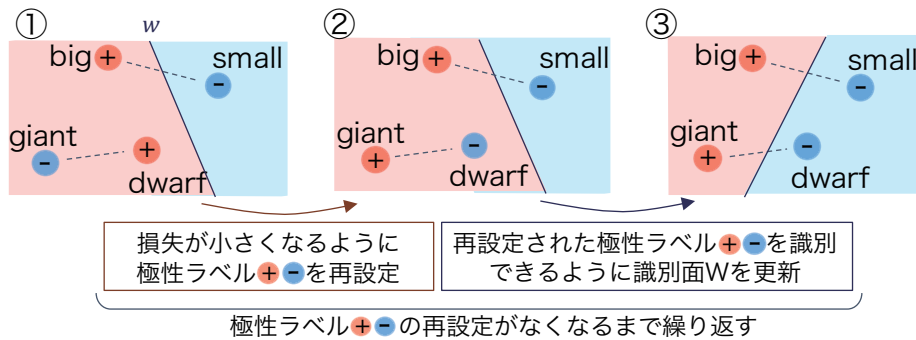


図 2: Transductive SVM を拡張した対義識別面の学習の概要。破線は対義の意味関係を表す。

アのように、感情極性では捉えられない対義軸（対義の視点）を持つものが多く存在すると予想される。そこで本稿では、図 1 のように、対義を表現する線形識別平面（以下、これらの識別面を「対義識別面」と呼ぶ）を複数用意し、

単語ベクトルが十分に似ていて、かつ対義識別面のどれかの極性が異なれば対義と判別し、全ての対義識別面で同じ極性を持たば同義と判別する単語埋め込み

を実現する前段として、その対義識別面を同義語・対義語からなる訓練事例から教師あり学習することを考える。ここで問題となるのは、様々な対義関係をどの対義識別面で表現すべきか、が訓練事例からはわからない点である。この問題に対して本稿では、学習と推論を交互に繰り返し行う **Transductive SVM** [Joachims 99] を拡張し、対義の分離軸（対義の視点）に対応する複数の対義識別面を同時に学習する手法を提案する。

2. TSVM を拡張した対義識別面の学習

本節では、WordNet[Fellbaum 98] のような資源から得られる対義・同義関係にある単語ペア集合を訓練データとして用いて、与えられた単語埋め込み空間内で対義を識別する対義識別面を学習する方法について説明する。

ただし本稿で対象とする対義極性は、感情極性におけるポジティブ・ネガティブのように単一の分離軸を設けることは難しいため、複数の分離軸に相当する複数の対義識別面を学習しながら、訓練データ中の対義語・同義語ペアに極性ラベルを設定する。本研究では、Transductive SVM (TSVM) [Joachims 99] のラベル設定及び識別面の学習方法を拡張して、学習と推論を交互に繰り返ししながら、対義語ペア・同義語ペアに対して極性ラベルを設定し、対義識別面を学習する手法を提案する。TSVM は、半教師あり学習の一つである Transductive 学習を SVM (Support Vector Machine) に適用し、ラベル付きの訓練データが少ない場合でも分類精度を向上させる手法で、学習と推論を交互に繰り返ししながら、学習が収束した時点の予測を最終出力とする。

TSVM を拡張して対義識別面を学習するタスク設定は、
入力 単語ペアの意味関係ラベル（対義・同義）、
単語埋め込みベクトル集合

出力 対義識別面（識別関数の係数ベクトル）

とし、訓練データの単語ペアの意味関係ラベル（対義・同義）を用いて、単語埋め込みベクトルに極性ラベル（+ または -）を設定しながら対義識別面の学習を行う。

提案手法は、1) 極性ラベルの設定ステップと 2) 対義識別面の更新ステップの 2 つのステップを極性ラベルの再設定が生じ

なくなるまで繰り返し行うことで、最終的な対義識別面を得る。学習の概要を図 2 に示す

2.1 極性ラベルの設定ステップ

ここでは対義識別面の数を 1 つと仮定し、学習における極性ラベルの設定ステップ（図 2 ①→②）について説明する。はじめに、訓練データ中の対義語ペアに対して一方の単語に正（+）の極性ラベル、もう一方の単語に負（-）の極性ラベルを設定する。その時、（+，-）と（-，+）の 2 通りの仮ラベルで識別損失を計算し、損失が小さくなる極性ラベルを設定する。例えば図 2 の①で、“giant” と “dwarf” のペアについて、対義識別面 w での識別損失を計算した場合、“giant” の誤分類による損失が大きいため、より損失が小さくなる極性ラベル（“giant” が +，“dwarf” が -）を設定する。この時、対義語ペアの各単語は必ず異なる極性ラベルとなる。同様に、同義語辞書の同義語ペアに対しては、両方の単語に正または負の同じ極性ラベルを設定する。その時、（+，+）と（-，-）の 2 通りの仮ラベルで分類誤りの損失を計算し、損失が小さくなる極性ラベルを設定する。

2.2 対義識別面の更新ステップ

次に、設定された極性ラベルを用いて、対義識別面 w を更新するステップについて説明する（図 2 ②→③）。対義語・同義語の識別損失関数 J として、ヒンジ損失関数を用いる。

$$J(w) = \frac{1}{N} \sum_{j=1}^N l(p_j, w), \quad (1)$$

ただし、 N は訓練データの単語ペア数、 p_j は訓練データの単語ペア、 $l(p_j)$ は p_j の対義識別面 w に関する損失を表す。 $l(p_j)$ は以下の式で計算される。

$$l(p_j, w) = \sum_{i \in p_j, w} \max(0, 1 - y_i(w \cdot v_i - b)). \quad (2)$$

y_i は前のステップで設定した極性ラベル、 w は識別関数の係数ベクトル、 v_i は単語埋め込みベクトル、 b はバイアス項を表す。

2.3 複数の対義識別面への拡張

ここまでの学習においては、対義識別面の数を 1 つと仮定していた。次に、複数の対義識別面を学習できるように極性ラベルの設定ステップと対義識別面の更新ステップを拡張する。

複数の対義識別面に拡張する前提として、図 1 の “hot” と “cold” のようにベクトル空間で近い位置にある対義は、全て

対義識別面 1	対義識別面 2	対義識別面 3	対義識別面 4
「明確性」	「平和的か」	「金権や物」	「方向」
blunt, sharp indefinite, definite stupid, intelligent cruel, kind	timid, bold refuse, offer war, peace unlawful, lawful	poverty, wealth new, second-hand natural, artificial king, subject	onward, backward internal, external sane, crazy impatient, patient

表 1: 各対義識別面で識別された対義語ペアの例 (4 対義識別面) 及び分離基準の解釈例

の対義識別面のうちのどれか 1 つの対義識別面で分離できれば対義であるとする。一方、同義については、全ての対義識別面で分離されなければ同義であるとし、この前提をみだすような拡張を考える。

複数の対義識別面を学習する場合、極性ラベルの設定ステップ (2.1 節) において、各対義識別面ごとに極性ラベルを設定する。対義識別面の更新ステップにおいては (2.2 節)、対義については、その対義識別面における損失が最小となる単語ペアを訓練事例として用いる。同義については、全ての単語ペアを訓練事例として用いる。これは、どれか 1 つの対義識別面でも分離されたら対義とする前提から、対義については最も識別しやすい単語ペアを分離できれば十分であり、一方同義については、全ての対義識別面で分離されなければ同義とする前提から、全単語ペアを分離しない制約が必要となることに対応した設計である。\$m\$ 個の対義識別面の集合を \$\mathbf{W}\$ とした時、対義識別面 \$\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m \in \mathbf{W}\$ に対して、対義語ペアの識別損失関数 \$J_a\$ を以下に定義する。

$$J_a(\mathbf{W}) = \frac{1}{N} \sum_{j=1}^N \min[l_1(p_j), l_2(p_j), \dots, l_m(p_j)], \quad (3)$$

ただしペア \$p_j\$ に関する対義識別面 \$\mathbf{w}_k\$ の識別損失を \$l_k(p_j)\$ とした。また、対義識別面の数 \$m\$ はハイパーパラメータである。同義語ペアの識別損失関数 \$J_s\$ を以下に定義する。

$$J_s(\mathbf{W}) = \frac{1}{N} \sum_{j=1}^N \sum_{k=1}^m l_k(p_j). \quad (4)$$

最終的に、対義識別面の更新ステップは以下の最小化問題として定義する。

$$\hat{\mathbf{W}} = \arg \min_{\mathbf{W}} \{J_a(\mathbf{W}) + J_s(\mathbf{W})\} \quad (5)$$

3. 実験

3.1 対義語・同義語の判定

学習された対義識別面の有効性を測るため、対義語ペアと同義語ペアからなる評価データを用い、性能評価実験を行う。評価時は、

入力 対義語ペアまたは同義語ペアの単語ベクトル
出力 ペアの各単語の極性ラベル

とし、1 対義識別面について、評価データの単語ペアに対し、異なる極性ラベルを予測すれば対義語ペア、同じ極性ラベルを予測すれば同義語ペアと判定する。複数の対義識別面では、全ての対義識別面の内、どれか 1 つの対義識別面でも対義を予測すれば対義語ペアとみなし、全ての対義識別面で同義と予測すれば同義語ペアとみなす。

3.2 実験設定

評価データとして、[Roth 14] [Nguyen 16] で使用されている対義語・同義語ペアのデータセット*1 から、形容詞・動詞・名詞の全ペアを合わせた 2100 ペアのうち、語彙内で対義・同義の訓練データ内のペアを除く 1864 ペア (対義 966 ペア、同義 893 ペア) を用いる。対義の訓練データには、WordNet[Fellbaum 98] 及び Roget's Thesaurus[Kipfer 09] から作成されたデータ*2 より、抽出したペア間の単語重複を除き、語彙内の 4996 ペア (9992 語) を用いる。同義の訓練データとして、WordNet [Fellbaum 98] から、対義の訓練データと単語が重複しない 3546 ペアを用いる。対義識別面学習に用いる単語埋め込みベクトルは、SGNS [Mikolov 13] モデルを用いて、英語 Wikipedia 上で訓練した 200 次元のベクトルを用いた。学習する対義識別面の数は 1 から 6 とする。

3.3 対義識別面による対義のクラスタリング

提案手法で学習された各対義識別面で識別された対義語ペアの例*3 を表 1 に示す。各対義識別面で分離された対義語ペアから、対義を分離する軸として、例えば対義識別面 1 は「明確性」、対義識別面 2 は「平和的か」、対義識別面 3 は「金権や物」、対義識別面 4 は「方向」のような基準で分けられていると解釈しうる。このように各対義識別面で識別されたペアから、どのような対義の軸が存在するか解釈の可能性があり、特徴量に用いた通常の単語ベクトルにも対義を識別する軸が存在しているのではないかと考えられる。各対義識別面において異なる性質の対義ペアが分離されることは、複数の対義識別面を学習するとき、各対義識別面において損失が最小となるペアを訓練事例に用いて対義識別面を更新するため、各対義識別面で別の分離軸が獲得されていくと考えられる。また、例えば対義識別面 4 で分けられた対義語では、「物理的な方向」と「性格的な方向」のように、対義識別面の数を増やすと、別の分離軸を獲得できる可能性がある。

3.4 対義語・同義語判定の性能

対義識別面の数と対義語・同義語の判定の正解率の関係を図 3 に示す。横軸は対義識別面の数、正解率を表す。対義識別面の数による正解率は、それぞれ対義と同義の両方の訓練データを用いて学習を行った場合 (左)、対義の訓練データのみを用いた場合 (中央) の正解率、チャンスレート (ランダムに対義または同義と出力したときの正解率) を示している。

図 3 から、対義識別面 1 つのとき、チャンスレートより正確に対義を見分けられており、TSVM を拡張した提案手法は

*1 http://www.ims.uni-stuttgart.de/forschung/ressourcen/experiment-daten/lexical_contrast_dataset.html

*2 <https://github.com/antonyms/AntonymPipeline/tree/master/data>

*3 対義の訓練データのみを用いて学習し、実験設定とは別の評価データ (対義の訓練データを取得したりポジトリ*2 の評価データ) での識別例を示している。

		予測				予測	
		対義	同義			対義	同義
正解	対義	556	410	正解	対義	805	161
	同義	298	600		同義	534	364

(a) 対義識別面の数=1

(b) 対義識別面の数=6

表 2: 対義語・同義語の判定の混同行列

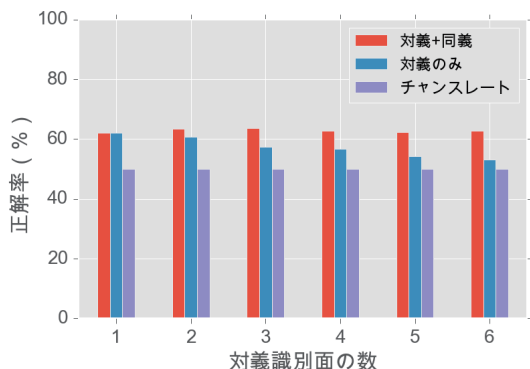


図 3: 対義識別面の数と対義語・同義語判定性能の関係

対義識別面の学習に効果があると考えられる。対義識別面の数を増やすと、対義の訓練データのみを用いて学習された対義識別面の正解率は低下した。これは、対義識別面の数の増加にともない、同義語ペアを対義識別面で分離してしまう確率が高まるのが理由としてあげられる。一方で、対義と同義の訓練データ両方を用いて学習された対義識別面の性能は、対義識別面の数に関わらず、一定を保つことができた。これは同義の訓練データを学習に用いることで、同義語を分離しない学習が行われた結果といえる。

対義と同義の訓練データ両方を用いて、1つの対義識別面、6つの対義識別面を学習したときの、対義語・同義語の判定における混同行列をそれぞれ表 2a, 表 2b に示す。対義識別面の数が1つのとき、対義の正解率は低く、それに比べ同義の正解率は高かった。対義識別面の数が6つのとき、対義の正解率は高く、それに比べ同義の正解率は低くなった。そのため、図 3 のように全体としての正解率は対義識別面の数を増やしても一定となっていることがわかる。対義を分離することと、同義を分離しないことは、概ねトレードオフの関係にある結果となっている。

4. 議論：複雑な同義・対義関係の学習

今回の実験設定では、同義の訓練データは、対義の訓練データと単語の重複がないペアのみを用いた。単語の重複がある場合、例えば図 4 における対義ペア “big”-“small” の “big” に対する “large” のような「対義語の同義語」を考える。ペアごとに関係をみだす極性ラベルを設定していくと、“big” と “large” の同義ペアに異なる極性ラベルが設定され、その極性ラベルを再設定して “big” の極性ラベルが変わると、“big”-“small” の対義ペアが同じ極性ラベルになることが起こりうる。対義・同義の関係には図 4 のような複雑な依存関係があり、両方の制約をみだすラベルを探索する必要がある。全単語の対義・同義両方の関係をみだす極性ラベルの探索は簡単ではなく、実験では訓練データから除いた。しかし、正しく対義を識別するためには、同義を分離しない対義識別面を学習する必要がある、

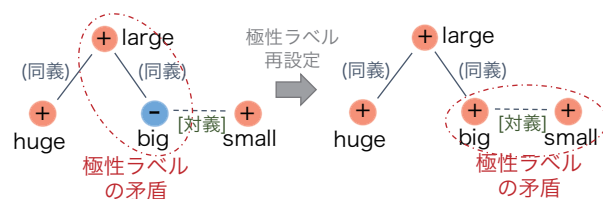


図 4: 複数の関係をもつ単語に対する極性ラベルの割り当て

対義語の同義語も同じ極性ラベルになるような考慮を要する。

5. おわりに

本論文では、単語埋め込み空間で対義の分離軸（対義の視点）に対応する複数の対義識別面を学習する手法を提案し、実際に学習した各対義識別面が異なる対義の分離軸を捉えることを示した。また、対義・同義の判定による評価から、対義と同義を正しく識別するためには、同義情報も利用した学習が有効であることを示した。一方で、より複雑な依存関係をもつ対義・同義の訓練事例を用いた対義識別面の学習における課題を論じた。

謝辞

本研究の一部は JST CREST (JPMJCR1301) および科研費 (15H01702) の支援を受けて行った。

参考文献

- [Chen 14] D. Chen, C. Manning: A fast and accurate dependency parser using neural networks, in *EMNLP*, pp. 740–750 (2014)
- [Faruqui 15] M. Faruqui *et al.*: Retrofitting Word Vectors to Semantic Lexicons, in *NAACL* (2015)
- [Fellbaum 98] C. Fellbaum: *WordNet: An Electronic Lexical Database*, Vol. 71 (1998)
- [Hill 15] F. Hill, R. Reichart, A. Korhonen: SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation, *Computational Linguistics*, Vol. 41, No. 4, pp. 665–695 (2015)
- [Joachims 99] T. Joachims: Transductive Inference for Text Classification using Support Vector Machines, in *ICML*, pp. 200–209 (1999)
- [Kipfer 09] B. Kipfer: *Roget's 21st Century Thesaurus*, Philip Lief Group (2009)
- [Melamud 16] O. Melamud *et al.*: The Role of Context Types and Dimensionality in Learning Word Embeddings, in *Proceedings of NAACL-HLT 2016*, pp. 1030–1040 (2016)
- [Mikolov 13] T. Mikolov *et al.*: Distributed representations of words and phrases and their compositionality, in *NIPS*, Vol. 2 (2013)
- [Mrksić 16] N. Mrksić *et al.*: Counter-fitting Word Vectors to Linguistic Constraints, in *NAACL-HLT*, pp. 142–148 (2016)
- [Nakamura 18] T. Nakamura, T. Ran, K. Inui: 単語の極性を埋め込んだ分散表現, 言語処理学会第 24 回年次大会予稿集 (2018)
- [Nguyen 16] K. A. Nguyen, S. Schulte Im Walde, N. T. Vu: Integrating Distributional Lexical Contrast into Word Embeddings for Antonym-Synonym Distinction, in *ACL*, pp. 454–459 (2016)
- [Nguyen 17] K. A. Nguyen, S. S. im Walde, N. T. Vu: Distinguishing Antonyms and Synonyms in a Pattern-based Neural Network, in *EACL*, Vol. 1, pp. 76–85 (2017)
- [Ono 15] M. Ono, M. Miwa, Y. Sasaki: Word Embedding-based Antonym Detection using Thesauri and Distributional Information, *NAACL* (2015)
- [Pham 15] N. T. Pham, A. Lazaridou, M. Baroni: A Multitask Objective to Inject Lexical Contrast into Distributional Semantics, in *ACL* (2015)
- [Roth 14] M. Roth, S. S. im Walde: Combining Word Patterns and Discourse Markers for Paradigmatic Relation Classification, in *ACL* (2014)
- [Rothe 16] S. Rothe, H. Schütze: Word Embedding Calculus in Meaningful Ultradense Subspaces, in *ACL* (2016)