

関係ベクトルを利用した皮肉の検出

肥合 智史 嶋田 和孝

九州工業大学大学院 情報工学府
九州工業大学大学院 情報工学研究院

{s_hiai, shimada}@pluto.ai.kyutech.ac.jp

1 はじめに

皮肉は、相手の欠点や弱点を意地悪く遠まわしに非難する表現である¹。例えば、

例 1 彼は部下に仕事を押し付ける素晴らしい上司だ。

という文において、書き手は「素晴らしい」という肯定表現を用いて、逆に、「上司」への非難を示している。このように、皮肉では、表層上の感情と、本当の意図が一致しないことがあり、皮肉の解釈には、表層だけではなく、文脈を考慮する必要がある。近年、より高度な文章理解のための挑戦的課題として、皮肉を検出する研究が進められている [1]。

本研究では、文脈情報の一つとして、皮肉の対象と対象に不満を持つ立場の関係に着目する。皮肉には、批判される対象が存在するという特徴がある [2]。また、皮肉には、期待に反したことへの否定的態度が表されているという指摘がある [3]。例 1 においては、「部下」の視点から、「上司」に対して、期待に反する働きぶりについて、皮肉が言われている。このような皮肉の対象と対象に不満を持つ立場の関係には、他にも、「生徒と教師」、「親と子」など、多様なパターンがあると考えられる。このような関係を表すベクトルを皮肉の検出に取り入れる。検出モデルには、リカレントニューラルネットワーク (RNN) を利用する。RNN によって、文に出現する単語の系列に加えて、皮肉の対象と対象に不満を持つ立場の関係を考慮した検出モデルを構築し、検出精度向上を目指す。

2 関連研究

先行研究において、皮肉の検出は、ある文が皮肉を含むかどうかの文書分類タスクとして取り組まれていることが多い。検出のための素性には、単語 N グラムや、ポジティブな表現とネガティブな状況を表す表現の連続 [6, 7]、単語の分散表現に基づく素性 [8] などが用いられている。

しかし、皮肉の検出には、表層情報だけでなく、文脈情報を考慮する必要がある。我々は、以前、立場表

現に着目した皮肉検出を行った [9]。立場表現は、例えば、例 1 中の、「部下」や「上司」といった立場を表す表現である。まず、皮肉に偏って出現する立場表現のリストを作成した。そして、その各表現が出現するかという二値素性を取り入れ、SVM により分類を行った。本稿でも、皮肉の対象と対象に不満を持つ立場を表す表現として、立場表現を利用する。ただし、本稿では、その表現それぞれではなく、立場表現のペアの関係を、文脈情報として利用する。立場表現のペアの関係は、分散表現の形で表される。

橋本ら [10] は、関係分類のための単語の分散表現の学習手法を提案した。橋本らは、二つの名詞の関係を表すベクトルを、その名詞それぞれや、その周り出現する単語の分散表現を組み合わせて構築した。本研究では、橋本らの手法を利用して、例 1 中の「上司」と「部下」のような、二つの立場表現の関係をベクトルで表現し、その関係ベクトルを検出モデルに取り入れる。

皮肉の検出モデルに、ニューラルネットワークを用いた研究が存在する [4, 5]。本研究でも、検出モデルに RNN を利用する。RNN に文中の単語の系列を入力し、それに続けて、二つの立場表現の間の関係を表すベクトルを入力することで、関係ベクトルを利用した皮肉の検出を行う。

3 提案手法

本節では、立場表現ペアの抽出と、立場表現ペアの関係を考慮した皮肉検出モデルについて説明する。

3.1 立場表現ペアの抽出

本節では、立場表現ペアの抽出について説明する。立場表現ペアは、(1) コーパスからの立場表現の抽出と、(2) 抽出された立場表現中での立場表現ペアの同定という二段階で抽出される。段階 (1) では、まず、「上司」、「部下」、「先生」、「生徒」などの単体の立場表現を抽出する。そのために、直前の語が立場であることを表す「にとっては」、「としては」という 2 つの手掛かり表現を利用する。コーパスから、各手掛かり

¹<https://www.weblio.jp/content/皮肉>

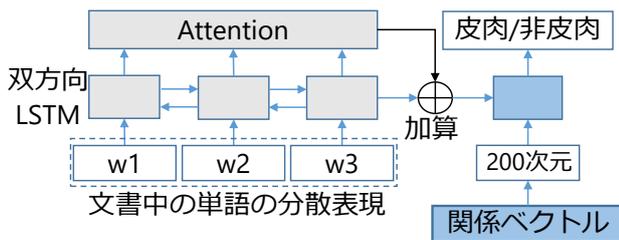


図 1: 提案手法の概要

表現の直前に出現する名詞を抽出し、立場表現とみなす。本研究では、コーパスには、Twitter への投稿を約 2300 万件ランダムで収集し、利用した。次に、段階 (2) において、立場表現の中から、「上司と部下」のような並列する立場表現の対を同定し、立場表現ペアとして抽出する。コーパス中で並列助詞の「と」の前後に、二つの立場表現が出現したとき、その二つを並列の立場とみなし、立場表現ペアとして抽出する。このときのコーパスにも、段階 (1) と同じデータを利用する。

3.2 RNN 分類器と関係ベクトル

本節では、提案手法において、皮肉を含むかどうかを分類するための、RNN モデルと、その入力について説明する。図 1 に本手法の概要を示す。

本手法では、RNN の一種である、双方向 LSTM [11] を用いる。双方向 LSTM のパラメータは、入力の次元は 200 次元、隠れ層の次元は 150 次元、エポック数は 30 に設定し、最適化手法には Adam [12] を利用する。この双方向 LSTM に、二種類の情報を入力する。一つは、文中の単語である。各単語は分散表現の形で入力される。もう一つは、立場表現ペアの関係を表すベクトルである。それぞれの入力について、詳しく説明する。

まず、文中の単語について説明する。双方向 LSTM に文中の単語を入力することによって、単語の系列を考慮できる。単語の系列は先行研究で用いられてきた素性の一つである。さらに、先行研究においては、肯定・否定語など、系列の中でも、検出に重要な語があることが確認されている [6, 9]。そこで、系列中の各単語に重み付けをして、考慮することができる Attention 機構 [13] を導入する。双方向 LSTM や Attention 機構は、先行研究にも用いられており、高精度な検出に寄与している [5]。単語系列の入力に対して、図 1 のように、Attention 機構を適用する。各単語は、word2vec [14] によって作成された 200 次元の分散表現の形で入力される。word2vec の学習には、3.1 節で使用した Twitter データを利用する。

次に、立場表現ペアの関係を表すベクトルについて説明する。3.1 節で抽出した、立場表現ペアのいずれ

かが文に含まれていれば、その関係を表すベクトルを双方向 LSTM に最後に入力する。この関係ベクトルは、橋本ら [10] の手法によって学習、構築される。橋本らは、関係分類タスクのために、名詞ペアの関係を表すベクトルを構築した。関係分類においては、関係を表したい名詞ペアだけではなく、その間や前後にどのような単語が出現するかという情報が必要である。例えば、

例 2 従業員や部下の仕事を上司がサポートする。

という文において、「部下」と「上司」の関係は、その間の、「仕事を」という部分や、その前後の「従業員や」、「がサポートする」といった部分に特徴づけられる。橋本らの手法では、関係を表したい名詞ペアを含む文について、名詞ペアを構成する二単語の分散表現と、その間・前後の単語の分散表現を組み合わせ、名詞ペアの関係を表す関係ベクトルを構築する。よって、各単語を表現する分散表現が必要になる。この分散表現は、大規模なコーパスから学習される。学習は、名詞ペアの間にどのような単語が出現するかを予測するモデルを基に行われ、関係を表したい名詞ペアを表現するために使われる分散表現 N と、その間や前後に出現する単語を表現するための分散表現 W がそれぞれ学習される。これを利用して、例 2 において、「部下」と「上司」の関係を表す場合、 N から「部下」、「上司」の分散表現が利用され、 W から「従業員」、「仕事」、「サポート」などの単語の分散表現が利用される。このように、対象の名詞ペアの分散表現と、その間・前後の単語の分散表現を組み合わせ、関係を表すベクトルが構築する。本手法では、橋本らの研究における名詞ペアを、立場表現ペアと置き換えて利用し、立場表現ペアの関係を表すベクトルを構築する。分散表現の作成や関係ベクトルの構築には、橋本らの実装²を利用する。学習のパラメータは、単語分散表現の次元は 100、学習率は 0.025、窓幅は 3、ネガティブサンプリングのサンプル数は 15 に設定した。また、学習のためのデータには、3.1 節で使用した Twitter データを利用する。

以上の方法で、各立場表現ペアに対応する 2000 次元の関係ベクトルを構築する。関係ベクトルでは、似た関係が似たベクトルで表現される³。この入力によって、皮肉に出現しやすい関係などの関係知識を考慮できる。ここで、文に複数の立場表現ペアが含まれていた場合、関係ベクトルの平均を入力する。また、文に立場表現ペアが含まれていなかった場合、関係ベクトルではなく、ゼロベクトルを入力する。

以上のように、双方向 LSTM に、文中の単語と立場表現ペアの関係を表すベクトルを入力する。ただし、各単語は 200 次元のベクトルで入力されるのに対して、関係ベクトルは 2000 次元のベクトルであるため、実

²<https://github.com/hassyGo/RelEmb>

³例えば、「上司・部下」の関係ベクトルと「社員・バイト」の関係ベクトルの類似度が高くなる。

装上の都合で、単語と同じように入力できない。よって、関係ベクトルを入力する際は、図1で示すように、関係ベクトルを200次元にする埋め込み層を追加して適用する。

4 実験

本節では、本手法の評価実験について説明する。評価用のデータセットをTwitterデータにより作成した。また、二つのベースライン手法を用意し、皮肉の検出精度を比較した。

4.1 データ

本研究では、日本語のTwitterへの投稿を対象に皮肉の検出を行う。多くの皮肉の検出の先行研究が、英語のTwitterへの投稿を対象にしている[1, 6, 7]。データセットの作成のためには、皮肉を含む投稿を収集しなければならない。この収集のために、多くの研究では、投稿者が投稿につけるハッシュタグが利用されている。ハッシュタグはその投稿の内容を表すタグである。英語の投稿においては、皮肉を表す「#sarcasm」というハッシュタグが付いた投稿を収集することで、皮肉を含む投稿が収集できる。よって、同様に、日本語の投稿で、「#皮肉」というタグが付けられている投稿を収集することを試みた。しかし、「#皮肉」というタグが付けられている投稿の数は十分ではなかった。そこで、「(皮肉)」という表現を利用した。日本語の投稿には、次の例のように、「(皮肉)」という表現を用いて皮肉であることを示すものがある。

例 3 彼は部下に仕事を押し付ける素晴らしい(皮肉)上司だ。

この「(皮肉)」という表現を含む投稿は、「#皮肉」というタグが付いた投稿より頻繁に投稿されていた。よって、「(皮肉)」という表現を含む投稿を21,000件収集し、皮肉の正例(皮肉ツイート)とみなした。この皮肉ツイートを、開発データに5,000件、訓練データに15,000件、テストデータに1,000件と分割して利用した。ただし、分類実験時は「(皮肉)」という表現自体が手掛かりにならないように、「(皮肉)」という表現は本文から削除した。また、開発・訓練・テストデータについて、皮肉ツイートと同数のツイートをランダムに抽出して、皮肉の負例(非皮肉ツイート)とみなして、データセットに加えた。

4.2 ベースライン

本手法の評価のために、先行研究のSVMを使った二手法をベースラインとした。また、本手法において、関係ベクトルを入力しない場合をベースラインとして、

分類実験を行い、精度を比較した。それぞれについて、詳しく説明する。

- ベースライン 1: 我々の以前の手法[9]を一つ目のベースラインとした。この手法では、皮肉を含む文により多く出現する立場表現のリストを利用する。そのリスト中のそれぞれの立場表現の存在を、SVMによる分類の素性に取り入れた。皮肉に、より多く出現する立場表現のリストを獲得するために、3.1節の段階(1)で抽出した立場表現リストを利用した。このリストから、開発データにおいて、非皮肉ツイートより皮肉ツイートに多く出現する立場表現を選び出した。BoWと、選び出された各立場表現の存在を素性としたSVMによる分類を行った。SVMの実装には、 SVM_{perf} [15]を利用した。この手法を、立場表現のペアを利用したRNNによる分類である、提案手法との比較対象の一つとする。
- ベースライン 2: Joshiら[8]の、SVMを利用した手法を二つ目のベースラインとした。Joshiらは、単語の1, 2, 3-gramsに加えて、単語分散表現を利用した素性を検出に取り入れた。この単語分散表現を利用した素性について説明する。まず、文中のすべての単語について、文中のその他の単語との分散表現同士の類似度を計算し、各単語について、最大・最小の類似度を確認する。そして、すべての単語の最大類似度、最小類似度同士を比較して、次の4つの類似度を素性に追加した。
 - 最大の類似度中の最大値
 - 最大の類似度中の最小値
 - 最小の類似度中の最大値
 - 最小の類似度中の最小値

SVMの実装には、ベースライン1と同様に、 SVM_{perf} を利用した。RNNを取り入れた本研究の提案手法でも、単語の分散表現を入力として用いるため、RNNとSVMを使った分類の比較のためにベースラインとした。

- ベースライン 3: Attention機構を取り入れた、双方向LSTMによる分類で、文中の単語分散表現のみを入力する場合の分類を三つ目のベースラインとする。提案手法では、関係ベクトルを追加して入力するが、関係ベクトルの有効性の評価のために、このベースラインを設定した。

4.3 結果

表1に分類実験の結果を示す。表中のP, R, Fはそれぞれ、適合率、再現率、F値を意味している。本研究の提案手法がすべての指標ですべてのベースライン

表 1: 結果

手法	P	R	F
ベースライン 1 (立場表現)	0.757	0.738	0.732
ベースライン 2 (分散表現)	0.771	0.755	0.751
ベースライン 3 (単語のみ)	0.773	0.773	0.772
提案手法	0.802	0.802	0.801

より高い精度となった。また、片側符号検定で有意差検定を行ったところ、提案手法は、すべてのベースラインに対し、 $p < 0.01$ で有意に高精度であった。また、ベースライン 3 は、ベースライン 1・2 に対し、 $p < 0.01$ で有意に高精度であった。ベースライン 3 が、ベースライン 1・2 より高精度であったことから、RNN を利用した分類モデルの有効性が確認された。また、提案手法が、ベースライン 2 より高精度であったことから、関係ベクトルの有効性が確認された。

5 考察

本手法では、3.2 節で説明したように、文中に立場表現ペアが含まれていれば、それに対応する関係ベクトルを入力し、含まれていなければ、その代りにゼロベクトルを入力する。4.1 節の開発データにおいて、立場表現ペアを含むデータの割合を確認したところ、皮肉ツイートで約 22%、非皮肉ツイートで約 12% であった。よって、関係ベクトルは、データの五分の一程度にしか適用されていない。

データの中には、3.1 節で抽出した立場表現ペアリストの網羅性の不足によって、関係ベクトルが適用されなかったものが存在すると考えられる。例えば、立場表現ペアリストに、「上司と部下」が含まれているが、その言い換えである、「上長と部下」は含まれていない場合、データ中に「上長と部下」が出てきても、立場表現ペアであると認識されない。

よって、今後はこのような言い換えを考慮した関係ベクトルの適用データの拡張について、有効性を確認していく。

6 おわりに

本研究では、皮肉によって批判される対象と、対象に不満を持つ立場の関係を考慮した、皮肉の検出手法を提案した。その関係を捉えるために、立場表現のペアの関係をベクトルで表現した。そして、RNN を用いた分類器にその関係ベクトルを入力し、皮肉の検出を行った。

評価においては、日本語の Twitter データを対象にした分類実験を行った。先行研究の SVM を利用した手法による分類と、関係ベクトルを利用しない RNN による分類をベースラインとして、分類精度を比較し、

提案手法の有効性を確認した。

謝辞：本研究の一部は科研費 17H01840 の助成を受けたものです。

参考文献

- [1] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, and A. Reyes. Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings 9th Int. Workshop on SemEval2015, Co-located with NAACL*, pp. 470–478, 2015.
- [2] J. D. Campbell and A. N. Katz. Are there necessary conditions for inducing a sense of sarcastic irony? *Language Resources and Evaluation*, Vol. 49, No. 6, pp. 459–480, 2012.
- [3] A. Utsumi. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. *Journal of Pragmatics*, Vol. 32, pp. 1777–1806, 2000.
- [4] A. Ghosh and T. Veale. Fracking sarcasm using neural network. In *Proceedings of NAACL-HLT*, pp. 161–169, San Diego, California, June 2016.
- [5] B. Felbo, A. Mislove, A. Sogaard, I. Rahwan, and S. Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on EMNLP*, pp. 1615–1625, Copenhagen, Denmark, September 2017.
- [6] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of EMNLP 2013*, pp. 704–714, 2013.
- [7] A. Joshi, V. Sharma, and P. Bhattacharyya. Harnessing context incongruity for sarcasm detection. In *Proceedings of ACL-IJCNLP*, pp. 757–762, 2015.
- [8] A. Joshi, V. Tripathi, K. Patel, P. Bhattacharyya, and M. J. Carman. Are word embedding-based features useful for sarcasm detection? In *Proceedings of EMNLP 2016*, pp. 1006–1011, 2016.
- [9] S. Hiai and K. Shimada. Sarcasm detection using features based on indicator and roles. In *Proceedings of the 3rd International Conference on SCDM 2018*, 2018.
- [10] K. Hashimoto, P. Stenetorp, M. Miwa, and Y. Tsuruoka. Task-oriented learning of word embeddings for semantic relation classification. In *Proceedings of the Nineteenth Conference on CoNLL*, pp. 268–278, Beijing, China, July 2015.
- [11] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, Vol. 45, pp. 2673–2681, 1997.
- [12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [13] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*, 2015.
- [14] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *The International Conference on Learning Representations: Workshops Track*, 2013.
- [15] T. Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD, KDD '06*, pp. 217–226, New York, NY, USA, 2006.