人工知能学会 インタラクティブ
情報アクセスと可視化マイニング研究会(第15回)
SIG-AM-15-01

分散表現を用いた有害表現判別に基づく炎上予測

Flame prediction based on harmful expression judgement using distributed representation

三宅 剛史[1*]　松本 和幸[1]　吉田 稔[1]　北 研二[1]
Takeshi Miyake[1], Kazuyuki Matsumoto[1], Minoru Yoshida[1], Kenji Kita[1]

[1] 徳島大学工学部知能情報工学科
[1] Department of Information Science and Intelligent Systems,Tokushima University

Abstract: In recent years, flaming on social media has been a problem. To avoid flaming, it is useful for the system to automatically check the sentences whether they include the expressions that are likely to trigger flaming or not before posting messages. In this research, we target two harmful expressions. There are insulting expressions and the expressions that are likely to cause a quarrel. Firstly, we constructed a harmful expression dictionary. Because a large cost requires to collect the expressions manually, we constructed the dictionary semi-automatically by using word distributed representations. The proposed method used distributed representations of the harmful expressions and general expressions as features, and constructed a classifier of harmful/general based on those features. An evaluation experiment found that the proposed method could extract harmful expressions with accuracy of approx. 70%. On the other hand, it was found that the proposed method could also extract unknown expressions, however, it wrongly extracted non-harmful expressions at a rate of approx. 40%.

1　はじめに

インターネットが発達したことにより SNS(Social Networking Service) やブログといったソーシャルメディアが広く利用されている．これらのサービスを利用することで，個人や企業，団体など誰でも自分の意見や広告などの情報を手軽に発信できるようになった．しかしその反面，発信した情報に対して多くの批判が寄せられるいわゆる「炎上」という現象が問題となっている．

ひとたび炎上が起こるとそのリスクは深刻である．個人情報の流出やそれによるメールや電話を使った問い合わせや嫌がらせ，人間関係の崩壊や全く関係のない人に影響が及ぶ場合もあり，炎上を起こしてしまった個人やその個人の所属する団体は社会的な信用を失ってしまうことになりかねない．こうしたリスクを回避するためにも，炎上を予防することが重要である．

炎上を予防するためには炎上についての理解が必要である．炎上にはさまざまな原因がある．よくあるケースとして，過激であったり，悪印象を与える内容でも SNS に対する理解が浅いため，安易に投稿してしまうことがある．例えばアルバイト先でのふざけた行動について写真付きで投稿したり，法律に触れるようなことを自慢（犯罪自慢）するような投稿である．これらの炎上を起こしてしまうユーザの多くが，なぜ自分の投稿が他の人に批判されたのか，どうして炎上してしまったのか理解できていないことが多い．そのため，第三者の視点で投稿内容を確認し，内容の指摘と訂正を促すことで炎上を未然に防ぐことができるのではないかと考えられる．

炎上事例を収集し，機械学習することにより炎上判別するという方法も考えられるが，炎上した発言そのものがすぐに削除されたりするため，データ収集が困難という問題がある．そのため，本研究では，収集しやすい悪口表現などを炎上予測の手がかりとする．

ある表現に対して炎上の可能性があるか否かの判断は難しいため，表現の単純な有害さだけで炎上を予測できるわけではない．しかし，表現が「有害である」と判断できるのであれば，その表現が原因で炎上の可能性があることを注意喚起することにより，炎上を予防できると考える．本研究では，有害表現を収集し，その分散表現を学習させ，機械学習により有害表現か否かを判別する分類器を作成することで炎上予防のためのシステムを構築することを目的とする．

*連絡先：徳島大学工学部知能情報工学科
〒 770-8506 徳島市南常三町 2 丁目 1 番地

2 関連研究

近年，炎上を工学分野において扱った研究は多く，様々なアプローチがある．山本ら [1] による，分散表現と連想情報を用いた道徳判断システムや，岩崎ら [2] による CGM における炎上の同定とその応用等がある．

本研究では，有害表現辞書の作成において，大量の有害表現を最初から人手により収集することはコスト面から不可能であると考えたため，人手により小規模な辞書を作成してから拡張する手法等を参考にする．

石坂ら [3] は電子掲示板「2 ちゃんねる」を対象として悪口表現の抽出をおこなっている．この研究では，掲示板上の悪口表現を自動抽出し辞書を作成することを目的としている．2 ちゃんねるから人手で表現を収集した小規模な悪口辞書を構築し，N-gram を用いた悪口表現の周辺単語モデルを作成し，このモデルを用いてさらに悪口表現を抽出することで辞書の拡張をおこなった．この手法では，新しい表現の抽出が可能であるが，悪口が書き込まれた掲示板の大規模なコーパスを用いなければ大幅な辞書の拡張は難しい．

また，既存研究では悪口表現の抽出をおこなっているが，本研究では差別表現や炎上しそうな単語も対象とする．差別表現は管理されるべき表現であり，また，炎上しそうな単語を抑制することは，炎上予防につながる．また，本研究では word2vec[4] による単語の分散意味表現を特徴として用いることで，コーパスさえ拡充すれば，類似した新しい有害表現の抽出が容易になる．

3 提案手法

3.1 提案手法の概要

図 1 に有害表現辞書の構築の流れを示す．まず Twitter やブログといったソーシャルメディアや書籍などのメディアから有害表現を抽出し小規模な有害表現辞書を作成する．つぎに，Twitter からランダムに収集したテキストコーパスを形態素解析器 MeCab[5] により分かち書きし，word2vec により分散表現を学習させる．有害表現辞書中の語の分散表現と類似する分散表現を持つ単語をコーパスから抽出し，有害表現辞書の自動拡張をおこなう．自動拡張する際に，意味的に類似しない語がノイズとして含まれる場合があるため，人手による選別をおこない，辞書を精錬していく．

図 2 に分散表現に基づく有害表現判定モデル構築の流れを示す．炎上判定のための分類器を作成するために，有害表現と対を成す表現として，一般的な表現を単語辞書等から収集し登録した一般表現辞書を作成する．有害表現辞書と一般表現辞書における単語の分散表現を素性とし，有害/一般クラスに分類するモデル（有害表現判定モデル）を Support Vector Machine(SVM) に学習させる．

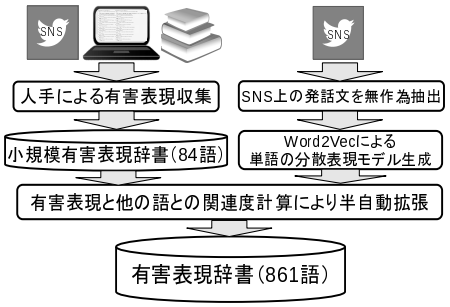

図 1: 有害表現辞書の構築

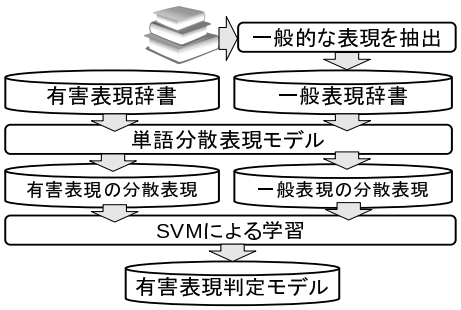

図 2: 分散表現に基づく有害表現判定モデルの構築

3.2 有害表現の抽出

3.2.1 有害表現の定義

本節では本研究での有害表現の定義について述べる．おおまかに 2 つの表現を対象とした．一つ目は悪口表現，差別表現等の人を罵ったり馬鹿にする際に使われる侮蔑表現，二つ目は政治や歴史問題，宗教等に関連するような意見が衝突しやすいデリケートな話題やタブーが存在するカテゴリに属する表現（以下，荒れそうな表現とする）を対象とした．表 1 に有害表現の一例を示す．

表 1: 有害表現の一例

表現名称	表現例
侮蔑表現	バカ　腰抜け　とろい 痴呆　ロンパリ　火病
荒れそうな表現	政治　宗教団体　戦争

侮蔑表現は誰もが使いがちであるが，炎上の可能性をもった表現の 1 つでありなるべく管理されるべき表

現である．悪口表現は誰に対して発言してもあまり好ましくない表現とし，差別表現は特定の属性を持った人たちに対して発言してはいけない表現と定義する．差別表現には，元々は差別表現ではなかった病名や症状等を表す表現を，他者を罵るために使い始めたことにより，新たな語義や用法が追加されたものがある．そのため，本来の意味で使用しているのか差別表現として使用しているのかを判別することは，単語単位だと困難であり，文脈等を考慮した語義曖昧性解消が必要になる．本研究で構築するシステムでは，疑わしい表現であれば，有害でない用法であっても炎上の可能性がある表現として提示する．

荒れそうな表現は，炎上しやすいカテゴリの表現を判定したいため本研究で扱う対象とした．意見の衝突しやすい話題では頻繁に口論が起き，そこから団体や企業，個人などを批判，誹謗中傷したりとエスカレートして状況が悪くなり，炎上に発展するケースが多くみられる．そのため，炎上しやすい話題に関する表現も有害表現として扱うことで，炎上を抑制できると考えた．

3.2.2 有害表現の抽出について

抽出における手順を述べる．大量の表現を人手により集めることが困難なため，人手で少量の表現を集めてそれを自動で拡張することを目指す．まず，Twitter，ニュースサイト，ブログ記事，掲示板，書籍等のメディアを対象として人手で有害表現を 84 単語収集し，小規模な有害表現辞書を作成した．収集する基準は 3.2.1 節において説明した有害表現の定義に従った．最近はあまり使われない古い表現も収集対象とした．

小規模な有害表現辞書と類似度の高い表現を収集するために，word2vec の学習をおこなう．まず Twitter から無作為に約 100 万ツイートを収集し，形態素解析により分かち書きしたものを用いて word2vec の学習をおこなった．word2vec の学習によりツイート内の単語をベクトル化することができ，ベクトルを比較することで関連度を調べることができる．word2vec の学習をおこなう際のパラメータは次元数を 200 とし，文脈の最大単語数である window 幅は 5，単語を無視する頻度である sample 値は 0.001 と初期設定のまま使用した．図 3 に，代表的な有害表現について分散表現ベクトルを t-SNE[6] を用いて次元圧縮し，二次元座標空間上にプロットしたグラフの一部を拡大したものを示す．

小規模な有害表現辞書の単語と学習した word2vec のモデルのベクトルの関連度を比較し，類似した表現をコーパスから自動抽出した．しかし，自動抽出した場合に，有害表現と意味は類似しているが有害ではない表現も収集してしまうため，抽出された語群に対し，人手により判断することで，有害な表現の選定をおこなった．3000 単語ほど収集したものから 861 単語を有害な表現とした．この 861 単語を登録した辞書を有害表現辞書とする．

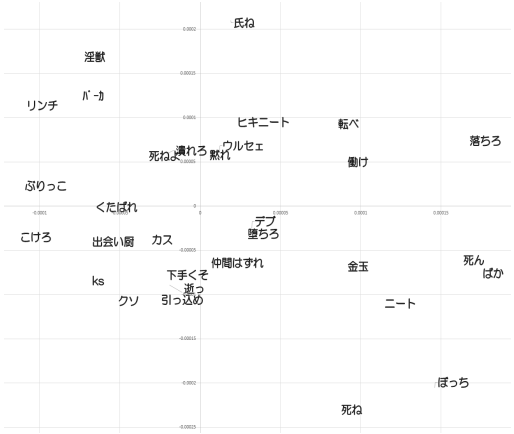

図 3: 有害表現の二次元座標空間へのマッピング

3.3 一般的な表現の抽出

3.3.1 一般的な表現の定義

一般的な表現とは，大衆に使われる有害でない表現とする．この表現の定義として「日本語の語彙特性 第 9 巻 [7]」を参照した．この書籍は日本語の単語に対して単語親密度という数値を表記しており，これは単語にどれだけなじみがあるかを 7 段階尺度 (1:なじみがない，7:なじみがある) で定義したものである．表 3 に単語親密度の例を示す．

表 2: 単語親密度の例

名称	単語親密度
星空	6.176
アーケード	5.688
偏光	3.824
あおざり	1.750

なじみがあるということは大衆に知られているということである．本研究では，親密度が高い単語かつ有害でない表現を一般的な表現とした．

本研究では有害表現を SVM に学習させるため，負例として反対の意味を持った有害でない言葉が必要である．そのため一般的な表現を収集して辞書を作成する．

3.3.2 一般的な表現の抽出について

一般表現の抽出について述べる．3.2.1 節で述べたように書籍をもとに表現を抽出する．日本語の単語に対

して 1 から 7 までの 7 段階尺度で親密度が表記されており，親密度が高いほどなじみがあるといえる．本研究では親密度が 6.3 以上の表現である 1384 語を抽出し，word2vec の学習において作成したモデルに含まれている単語で有害な表現ではないものを選定し，そこから無作為に 900 語を選出した．この 900 語を登録したものを一般表現辞書とする．表 3 に抽出した表現の例を示す．

表 3: 一般的な表現の例

名称	単語親密度
アイスクリーム	6.562
正直	6.375
うるさい	6.531
会う	6.594

3.4 炎上判定システムの作成

3.4.1 分類器の作成

3.2 節と 3.3 節で収集した有害表現辞書と一般表現辞書を用いて SVM による学習をおこなう．SVM による学習をおこなうためには素性ベクトルを学習させる必要がある．学習用データとして有害表現 864 語，一般表現 900 語を用いる．これらを word2vec の分散表現モデルを参照し，分散表現に変換したものを素性ベクトルとして用い，SVM による二値（有害/一般）分類器を学習し，有害表現判定モデルを作成した．SVM の学習にはデフォルトパラメータを使用し，有害表現を正例，一般表現を負例とした．

3.4.2 有害表現判定モデルの検証

構築した有害表現判定モデルの有効性を確認するため，10 分割交差検証をおこなった．交差検証の結果（精度，$\pm 2\sigma$ レンジ）を表 4 に示す．$\pm 2\sigma$ は標準偏差を 2 倍したもので結果のばらつきを表している．このことから分類器の精度は高く，かつ結果のばらつきは極めて少ないことがわかる．

表 4: 10 分割交差検証の結果

平均精度	$\pm 2\sigma$ レンジ
0.96	(+/- 0.03)

3.4.3 有害表現判定システムの作成

作成した分類器を用いて炎上の可能性を判定するシステム（有害表現判定システム）を作成する．入力された文章に対して形態素解析器により単語単位に分割し，各単語に対して word2vec で学習した分散表現のモデルを参照する．モデル中に単語が存在する場合，分散表現に変換して分類器に入力することで，炎上の可能性がある表現が含まれているかを判定して表示する．システムの画面例を図 4 に示す．

図 4: 有害表現判定システムの Web インタフェース

このシステムは文章中に含まれる単語を有害かどうか判定し，どの単語が炎上の可能性があるかを表示する．表現が有害なら炎上するとは限らないが，炎上の要因として暴言や悪口などの有害な表現は存在する．そのため訂正を促し炎上の可能性を抑制するねらいがある．

4 評価実験

4.1 実験データ

本研究で使用する実験データは，実際にインターネット上で問題となっている悪口文を用いる．サンリオのキャラクターである「シナモン」の公式ツイート [8] に対し，多数の悪口文が投稿されている．いじり感覚の悪ふざけから発言が過激化しネット上でのいじめに発展している．いじめの対象が一個人ではない実在しないキャラクタということもあり，投稿が削除されたり，アカウントが削除される事態には至っていない．この

多数の悪口文を実験対象として用いることで炎上予測性能を評価できると考えた．

実験データについて表 5 に示す．悪口文は，2015 年に投稿されたツイートから計 108 文を収集した．一部，悪口文とは言えない文もあるが，誤検出の確認のために残している．表 6 に悪口文を抜粋したものを示す．これらの悪口文を人手により判断し，システムが有害と判定すべき 35 単語の表現を正解表現とした．正解表現の一覧を表 7 に示す．

表 5: 実験データ

データ名	数量
シナモンへの悪口文	108 文
正解表現	35 単語

表 6: シナモンへの悪口文の例

悪口文	正解表現
ブロック解除しろ事故って死ね	死ね
雲から落ちろ	落ちろ
ええかげんぶりっこ卒業して	ぶりっこ

表 7: 正解表現

正解表現一覧			
死ね	転べ	落ちろ	死ん
仲間はずれ	ぼっち	金玉	働け
堕ちろ	ks	黙れ	淫獣
バ ーカ	ばか	ぶりっこ	やろう
消せ	転ん	ヘイト	出会い厨
こけろ	氏ね	ウルセェ	下手くそ
引っ込め	クソ	ニート	カス
くたばれ	潰れろ	リンチ	デブ
ヒキニート	逝っ	死ねよ	

4.2 評価方法

悪口文からシステムに判定してほしい有害表現を正解表現として選定する．この選定基準は有害表現辞書を作成する際の基準と同じである．正解表現を 35 単語選定し，これらをシステムが有害と判定できればシステムは正常に判定できたとする．

正解表現が判定できているかの再現率および適合率と，誤って有害でない表現が判定されてしまった誤検出率などを比較する．また，SVM を用いずに辞書のみを用いたマッチングで有害表現がどれだけ判定できるかを実験し，SVM の有用性についても調査する．

4.3 実験結果

実験結果を表 8 に示す．

表 8: 実験結果

	提案手法	辞書のみ
判定成功	26	10
誤検出	20	0
再現率	74.3	28.6
適合率	56.6	100

この表から，SVM を用いることで有害表現辞書に存在しない未知の有害表現の判定が可能なことがわかる．また，当然ながら，正解表現のうち有害表現辞書に含まれる表現については，すべて有害表現として判定できた．

実験において判定できた表現等について表 9 に示す．誤検出をした表現について表 10 に示す．

表 9: 判定に成功した表現と失敗した表現の内訳

表現一覧			
成功表現		失敗表現	辞書
死ね	ばか	死ん	ばか
転べ	ヘイト	金玉	ヘイト
落ちろ	ぼっち	働け	ぼっち
仲間はずれ	クソ	やろう	クソ
バ ーカ	ニート	消せ	ニート
堕ちろ	カス	転ん	カス
黙れ	デブ	下手くそ	デブ
淫獣	氏ね	引っ込め	氏ね
こけろ	くたばれ	逝っ	くたばれ
潰れろ	ヒキニート		ヒキニート
ウルセェ	ぶりっこ		
ks	出会い厨		
死ねよ	リンチ		
26 単語		9 単語	10 単語

表 10: 誤検出された表現

「w」の羅列・記号	怪我・病気	命令語	その他
w	植物状態	どけよ	危
www	粉砕骨折	働けよ	非常識
wwww	永眠	蹴れ	
wwwww	骸骨	浮け	
w*25		染めろ	
！ *9			

この表を見ると「w」の羅列を有害と判定しているパターンが多い．病気や怪我を表す語も含まれていることがわかる．また，命令語が含まれている．

5 考察

SVM による正解表現の判定が 26 単語と，有害表現辞書による単純なマッチングよりも多く判定できている．この結果から，有害表現の判定において，分散表現を素性とした SVM による分類が有効であることが分かる．学習させた有害表現に近い意味の単語を有害表現として判定できたと考えられる．一般的な表現 20 単語が有害表現と判定された．表 10 から，「w」の羅列が多いことが分かる．分散表現の学習段階において，「w」が有害と考えられる文章に頻出していたことが原因と考えられる．「w」はネットスラングであり，人を嘲笑ったり，小馬鹿にする際によく用いられる．

「!!!!!!!!」も同じようなケースで，命令文や罵倒と一緒に用いられることが多いためだと考えられる．また，侮蔑表現に病気を基にした表現が多いことから，侮蔑表現と，怪我や病気を表す表現の分散表現が類似してしまい，有害表現として判定されてしまったと考えられる．また，同様に，命令語が，罵倒や悪口などと分散表現間の類似度が高くなる傾向にあったため，システムが有害と判定してしまった．

6 まとめ

本研究では，ソーシャルメディアや書籍等から有害表現を収集し有害表現辞書を作成した．本研究での有害表現とは悪口表現，差別表現，荒れそうな表現のことである．Twitter から 100 万ツイートを取得し，word2vec による学習をおこない有害表現と類似する語をコーパスから抽出することで，有害表現辞書の拡張をおこなった．

有害表現と対を成す表現として一般的かつ有害でない表現を収集し，一般表現辞書を作成した．有害表現辞書と一般的表現辞書を学習データとして，SVM による学習をおこない，有害表現を判定する有害表現判定モデルを作成した．このモデルを用いて，有害表現判定システムを作成し，悪口文を用いた実験をおこなった．結果として，辞書のみを用いた手法よりも多くの有害表現を抽出可能なことが分かった．誤って判定されてしまった表現について分析した結果，類似語の影響や用法に依存して有害とも解釈できてしまう語が存在することが分かった．今後の課題として，有害表現辞書の拡充と，用言や複数の語に分割されてしまう表現，文脈によって有害と判断される語への対応が考えられる．

謝辞

本研究の一部は，科学研究費補助金 15K16077, 15K00425, 15K00309 によりおこなわれた．

参考文献

[1] 山本 眞大, 萩原 将文: 分散表現と連想情報を用いた道徳判断システム, 日本感性工学会論文誌, Vol.15, No.4, pp. 493–501, 2016.

[2] 岩崎 祐貴，折原 良平，清 雄一，中川 博之，田原 康之，大須賀 昭彦: CGM における炎上の同定とその応用,The 27th Annual Conference of the Japanese Society for Artificial Intelligence, 2013.

[3] 石坂 達也，山本 和英: 2 ちゃんねるを対象とした悪口表現の抽出，言語処理学会第 16 回年次大会, pp.178–181, 2010-03.

[4] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, CoRR, abs/1310.4546, 2013.

[5] MeCab: Yet Another Part-of-Speech and Morphological Analyzer http://taku910.github.io/mecab/.

[6] Maaten, L., Hinton, G.: Visualizing Data using t-SNE，Journal of Machine Learning Research, Vol.9, pp. 2579–2605, 2008.

[7] 天野 成昭，近藤 公久，笠原 要，NTT コミュニケーション科学基礎研究所（監修），NTT データベースシリーズ 日本語の語彙特性〈第 9 巻〉単語親密度 増補，三省堂，2008.

[8] シナモン【公式】Twitter アカウント： https://twitter.com/cinnamon_sanrio?lang=ja