情報処理学会論文誌 Vol.59 No.2 429–441 (Feb. 2018)

分類手法に応じた他者を怒らせる問題発言抽出パターンの特徴に関する分析

村山 大騎[1,a)] 宇田 隆哉[1,b)]

受付日 2017年5月1日, 採録日 2017年11月7日

概要：SNS の流行にともない，グループや不特定多数に向けて個人が自由に意見を述べられるようになってきている．SNS では頻繁に投稿が行われるようになったことから，不注意で不用意な発言をしてしまい，しばしばトラブルが発生する．SNS の利用者にとって，トラブルが発生した後にそのトラブルの原因となった発言を特定することはそれほど困難ではないが，トラブルが発生する前にその発言がトラブルを発生させうるかどうか，投稿者が 1 人で判断することは困難である．つまり，それぞれの発言は，投稿前に客観的に評価されていない．そこで，本論文では，過去の発言から深層学習を用いて評価モデルを作成することで，これから発言しようとしている内容が，過去にトラブルを引き起こした発言と類似しているかを評価して提示する手法を提案する．提案手法を用いれば，投稿前にその内容について注意を促せる可能性がある．

キーワード：SNS，メッセージフィルタリング，深層学習

Analysis in Classification of Patterns with Characteristics Extracted from Controversial Statements with Anger of Others

Daiki Murayama[1,a)] Ryuya Uda[1,b)]

Received: May 1, 2017, Accepted: November 7, 2017

Abstract: Popularity of SNS enables people to open their opinions to groups or the general public. Frequent posts often make troubles since people sometimes do not carefully confirm them before posting. It is not quite difficult to find out the reason of a trouble after the occasion of the trouble in SNS. On the other hand, it is difficult for a poster to predict whether his or her message makes a trouble or not. This is because any message do not objectively evaluated before posting. Therefore, in this paper, we propose a method to indicate whether the message is similar to messages which made a trouble or not by evaluating with evaluation models which are created from past posts by deep learning. Our method can make posters be aware their messages which should be confirmed.

Keywords: SNS, message filtering, deep learning

1. 序論

SNS の流行にともない，インターネットを通して個人が意見を世界中に発信できるようになった．しかし，不用意な発言はときに他者の反感を買い，発言者への誹謗中傷が殺到する場合もある [1]．これを防ぐには，投稿される前にその発言内容がトラブルにつながるかどうか，客観的な視点で確認する必要がある．本論文では，投稿されようとしている発言が，過去に他人を怒らせた発言と類似しているかどうか確認する．提案手法を用いれば，投稿前にその内容について注意を促せる可能性がある．具体的な使用方法としては，SNS での発言時に，「この発言は過去に他人を怒らせた発言と類似していますが再考しますか？」のように問い，利用者に「このまま投稿する」や「キャンセルし

[1] 東京工科大学
Tokyo University of Technology, Hachioji, Tokyo 192–0982, Japan
[a)] c0113509c0@edu.teu.ac.jp
[b)] uda@stf.teu.ac.jp

© 2018 Information Processing Society of Japan

て再考する」を選択させられることを検討している．このようにすれば，利用者はもう 1 度冷静に発言内容を確認したり，発言自体をとりやめたりできる．

人の感情を分析する研究には様々なものがある．記述内容から感情を分析するためには感情を分析するモデルが必要である．モデルとしては，ナレッジベースモデルや機械学習ベースのモデル，Lexicon ベースのモデルなどが存在する [2]．

ナレッジベースモデルとは，特定の単語を人間が考慮したルールに基づくアルゴリズムによって定義するモデルである．これに対し，機械学習ベースのモデルでは，データをもとにして機械がモデルを作成し，人間はルールに関与しない．ナレッジベースモデルの場合には大量のデータは不要であるが，機械学習ベースのモデルでは，精度を高めるために一般的に大量のデータが必要となる．感情分析精度に関しては，大量のデータさえあれば機械学習ベースのモデルのほうが高いことが示されている．一方，ナレッジベースのモデルは人間がアルゴリズムを考えているため，アルゴリズムに抜けや漏れが生じやすいとされている．

機械学習の手法としては，Support Vector Machine（以下, SVM）や Naive Bayes（以下, NB），Convolutinal Neural Network（以下，CNN）と呼ばれるものがある．これらを用いて，記述内容から得られる人間の感情をポジティブ，ネガティブ，それ以外のように分類する学習を行うことによって，未知の記述内容の判別が行われる．なかでも CNN を用いた研究において，記述内容から感情を高い精度で分類することに成功している．実際に Twitter Sentiment Analysis という Twitter を用いた感情分析では，映画のレビューや顧客のレビューなどにおいて，SVM や NB よりも高い精度を出すことに成功している．

本研究では，読者の怒りの感情を正確に判別するために，CNN と Multinomial Naive Bayes（以下，MNB）を組み合わせた 2 段フィルタを用いる手法を提案する．本研究は，既存研究とは異なり，記述内容から記述者の感情を分析するものではないが，感情を分類するという点では既存研究と同じであるため，既存研究で高い精度を誇る CNN を利用している．さらに，本研究では，CNN を用いた際の致命的な誤分類を，MNB を追加した 2 段フィルタを用いることによって訂正できるように工夫している．

2. 関連研究

本章では機械学習を用いた感情分析の関連研究について述べる．

2.1 SVM による感情の分析

Mohammad は，Twitter のハッシュタグを用いてコーパスを作成し [3]，SVM を用いたモデルで Ekman が提唱した 6 つの基本的な感情（喜び，悲しみ，怒り，恐れ，嫌悪感，驚き）[4] を抽出した．Mohammad はそれぞれの感情を表すハッシュタグからデータをラベル付けした．その結果，本研究とも関係のある，怒りの感情に対する F 値は 34.5%であり，他の感情に対する F 値よりも低く，怒りの感情を抽出することは難しいことが分かる．

また，Roberts らも同様に Ekman の提唱した 6 つの基本的な感情を用いた分類を SVM を用いて行っている [5]．Roberts らの分類では，怒りの感情に対する F 値は 64.2%となり，Mohammad が導出した F 値（34.5%）よりも良い結果を出すことができた．

Mohammad や Roberts らの研究において，怒りの感情の分類がうまくいかなかった理由としては，怒りの感情を抽出したサンプルが他の感情のサンプルよりも少なかったことがあげられる．機械学習の精度を上げるには，一般的に大量のデータが必要だからである．

一方，Sintosova らは，Twitter を用いた感情認識のための半自動的な学習方法を提案した [6]．Sintosova らは，一般に使われている絵文字をもとにしたサンプルを収集し，重み付けされた SVM を作成することによって記述自体から感情を検出した．このモデルは，彼らが分類を行うスポーツ分野の記述のみを学習，テストするために用いている．このモデルは NB をもとにしたモデルよりもより良い結果を出すことに成功した．

S. Kim らは Plutchik が提唱した感情モデル [7] を参考にした 9 種類の感情（喜び，怒り，驚き，恐れ，悲しみなど）をもとにした感情モデルを作成した [8]．S. Kim らは，それらの感情が自然言語によってどのように作用するのかを分析した．彼らはモデルに SVM を使用しており，ポジティブな感情からネガティブな感情までのうち，会話した相手の感情に影響を与えたトピックには，心配とからかいと不満が含まれていたと分析している．S. Kim らは，会話の相手の感情に影響を与えた原因を分析し，特定しようとしているが，我々の研究においては，その原因が何であるかを分析することなく，同一または類似の特徴を持つ記述内容を，畳み込みによる処理で分類する．

2.2 CNN による感情の分析

Neural Network を用いた研究の中でも，CNN を用いたものが成果をあげている．CNN を用いた研究として，Twitter を用いた感情分析（ポジティブかネガティブに分類する）を行ったものがある．Y. Kim は，単純な構造の CNN を用いたモデルを用い，1 つの文章に関して，ポジティブかネガティブかの感情分析を行った [9]．彼が提唱したモデルは，画像で用いられるようなモデルとは異なり，層を少なくしたものである．彼のモデルは，映画のレビューに関するタスクや顧客のレビューに関するタスクなどで他のモデル（Recurrent Neural Network，NB，SVM，他の CNN）よりも高い精度を出すことに成功している．

© 2018 Information Processing Society of Japan

また，Kalchbrenner らは，文章の意味を分析する動的な CNN を提案した [10]．Kalchbrenner らのネットワークでは，それぞれの文章に対してすべての文章に適用される動的な k-最大プーリングを利用している．このモデルは，Y. Kim のモデルよりも少し複雑であり，このモデルを用いて様々なタスクが試された．彼らのモデルは，映画のレビューの複数クラスへの感情分析や，質疑応答や Twitter を用いた感情分析において，他のモデルと比較して最も高い精度を出すことに成功した．

Y. Kim や Kalchbrenner らによる既存技術の感情分析と，我々の研究には根本的な違いがある．既存技術においては，文章の記述内容を分析し，その記述を行った記述者がどのような感情を持ってその記述を行ったかを調べているが，我々の研究では，その記述を読んだ読者がどのような感情を持つかである．つまり，記述者が愉快であっても読者は怒りがこみあげている場合があるということである．

3. 提案手法

3.1 提案概要

SNS を用いて誰もが自由に意見を発信できるようになった一方で，発言が原因となり他人を怒らせてしまうことがある．そこで我々は，CNN と MNB を組み合わせた 2 段フィルタを用いて，他者を怒らせる問題発言を分類する手法を提案する．本手法では，Y. Kim の実装に類似したシンプルな CNN と，従来の様々な分類に利用されている MNB を組み合わせた 2 段フィルタを用いる．

人を怒らせる発言のサンプルは Twitter を用いて収集した．ある発言が人を怒らせたかどうかの判断は，我々の予想によるものではなく，実際に他者が怒った結果をハッシュタグを用いて自動的に収集し，その元となる発言を抽出している．

Twitter の 1 つの発言に含まれる単語をそれぞれベクトルに変換し，そのベクトルの集合を 1 つのサンプルとした．このサンプルを CNN を用いて学習と分類，また MNB を用いて分類を行っている．

手順は次のとおりである．まず，サンプルを学習用とテスト用に分ける．学習用サンプルを用いて，CNN の学習を行い，CNN のモデルを作成する．出力は，人を怒らせるかどうかの 2 値分類であり，これを CNN のフィルタとする．同様に，学習用サンプルを用いて，MNB のフィルタも作成する．評価の際には，テスト用サンプルを，まず CNN のフィルタを通し，人を怒らせると分類されたものを抽出，人を怒らせないと分類されたものを MNB のフィルタに通す．MNB のフィルタ通過後に，人を怒らせないサンプルは問題ないことになり，それ以外のサンプルは問題発言となる．

2 段フィルタを用いている理由は，CNN でうまく畳み込めない例があるからである．2 段フィルタを用いることにより，CNN で発見できなかった問題発言も，MNB で拾うことができる．

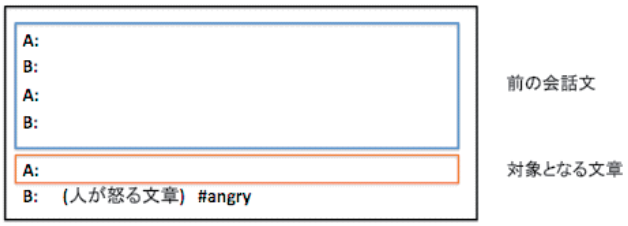

図 1 本研究が対象とする記述
Fig. 1 Objective description of our research.

3.2 データ収集

本研究では，SNS の知名度，不特定多数に向けて発信される点を考慮して，Twitter の投稿を識別の対象とした．なお，収集した投稿は，研究のグローバル性を考慮し，英語のもののみとした．**図 1** に，本研究が収集の対象とする記述について示す．

A と B は Twitter で発言を行っているユーザを表している．この例では，図の一番下に示される B が怒りを感じており，その直前の A の記述が我々が評価の対象とする文章となる．また，評価の対象となる文章の前にある会話文も学習用に収集する．なお，B が怒りを感じた最後の B の記述は使用しない．つまり，これがこの一連の会話の結論であり，これを学習や評価を行う際のラベルとするのである．具体的には，Twitter 上で #angry，#fuck，#bitch のハッシュタグが付いたものを，人を怒らせる文章とした．逆に，人を怒らせない文章の収集に関しては，人が怒るハッシュタグ以外のものを起点に会話の始めまでをたどることによって収集した．

Twitter から人を怒らせると想定されるツイートを収集する際，いくつかのハッシュタグを用いた理由は，類似の手法を用いている先行研究があったためである．Twitter から怒りのツイートを収集するにあたり，Dini らは angry という単語を利用している [11]．また，Roberts らの研究 [5] および Mohammad の研究 [3] においては，怒りのツイートの収集に anger という単語を利用している．

これらをふまえ，本研究においては #angry というハッシュタグが付与されているツイートを怒りに関するものとして収集することとした．なお，#anger のハッシュタグを用いなかった理由は後述するが，収集できたサンプルの数が少なかったためである．

また，著者らの主観に基づいてはいるが，人を罵る代表的な言葉を他にも列挙して調査した．著者らが思いついたものは fuck，bitch，bullshit，holy shit，asshole，cunt である．なお，「Mind your own business」のように単語数が多い言葉に関しては，ハッシュタグにされにくいと判断し調査対象から除外した．

© 2018 Information Processing Society of Japan

これらのハッシュタグを用いているツイートから，それぞれ 10 サンプル程度をランダムに選択し，これらの大部分が本当に怒りのツイートであるかどうか確認した．怒りのツイートであるという根拠を機械的に分類することは非常に困難であるため，著者らの主観となるが，それらが冗談などではなく，明らかに怒りが込められていることを確認した．その結果，angry のほか，先述の fuck，bitch，bullshit，holy shit，asshole，cunt は，すべて怒りの内容をともなうハッシュタグであることが確認できた．なお，明らかに怒りが込められているかどうか確認する際，文面だけでなく “!” の数や怒っている絵文字も判断材料とした．“!” が 3 つ以上含まれているもの，怒っている絵文字が 1 つ以上含まれているものは，明らかに怒りが込められているものとした．

その後，Twitter API を用いて対象のハッシュタグが付与されているツイートを収集したうえで，収集できたサンプルの数を比較した．ここで，#angry が付与されたものは 106 個，#fuck のものは 136 個，#bitch のものは 385 個であり，それ以外のものは 30〜40 個にとどまった．本研究においては，リプライ（返信）形式になっているツイートでないと利用できない．リプライ形式になっていれば，元のツイートに対して怒りを感じていることになる．もちろん，自分と同意見の他人の怒りのツイートに対して，賛同という意味でリプライ形式を使用することも可能ではあり，また，無意味にリプライ形式を使用することも技術的には可能であるため，絶対にそうであるとはいいきれないが，先述の 10 サンプル程度の調査の結果，そのようなものは見つからなかったため，存在したとしても非常に少数であり本研究においては無視できるものと判断した．

一方，リプライ形式になっていないツイートの場合には，何に対して怒りを感じているかが明確でないため，本研究の対象からは除外した．もちろん，怒りを感じたツイートに対して，リプライ形式以外で怒りを込めたツイートをすることは可能である．しかし，この場合には元のツイートを機械的に特定することは困難である．以上より，本研究が分類の対象とするものは，怒りに関するハッシュタグが付与されていて，なおかつリプライ形式であるツイートのみとした．また，非常に稀ではあるが，英語のみのツイートを収集した中にスペイン語のものなどが混在していた．言語の選別は Twitter の設定で行っているが，おそらく英語を含む 2 カ国語以上を扱える者によるツイートと思われる．これらのツイートは本研究の対象から除外した．言語が英語かどうかについては，Twitter API の lang が “en” となっているかどうかで判断した．

以上を考慮し，怒りに関するハッシュタグが付与されており，リプライ形式を含み，英語であるツイートに絞った場合，#angry のものは 66 個，#fuck のものは 84 個，#bitch のものは 235 個となった．よって，これらの 3 つのハッシュタグが付与されたツイートを本論文の分類対象とした．これらからランダムに選択した 300 サンプルを，兆候ありのものとして学習および評価に使用している．選択された 300 サンプルは，同一のものを本研究のすべての学習および評価に使用した．これら以外のハッシュタグに関しては，10 個程度のサンプルになってしまうため，分類の対象から除外した．たとえば，10 個である場合，そのうち 1 つが分類できるかどうかで 10%精度が変わってしまうためである．

テストデータのサンプル数 600 個は Twitter の API である REST API を用いて収集した．この API の制限として，過去 7 日間以上遡ってツイートを取得することができない．本研究で使用したサンプルは，2016 年 10 月 7 日から 10 月 13 日の間のツイートをこの API を使用して収集したものであり，兆候（人を怒らせる可能性のある記述）ありの 300 個と兆候なしの 300 個で合計 600 個となった．なお，継続的にデータを収集してサンプルを増やすことは可能である．

3.3 データのラベル付け

サンプルの分類を行うために，それぞれのサンプルに対してラベル付けを行う．本研究では，人が怒る前兆が文章にあるかどうかを判別しており，2 値に分類する．ラベル付けの例を**表 1** に示す．

表 1 では，読み手を怒らせた文章の前の会話文（図 1 の青い矩形で囲った部分）に対しては ‘1’ を，怒らせなかった文章の前の会話文（図 1 の青い矩形で囲った部分に相当するが，文章自体は人を怒らせていないもの）に対しては ‘0’ をラベル付けした．ラベル付けについては，4.2 節で示したように，#angry，#fuck，#bitch のハッシュタグが付いているかどうかで判断している．

3.4 文章の前処理

Twitter から文章を収集した後，その文章からベクトルを作成する前に前処理を行う．具体的には，URL や # が入ったハッシュタグを取り除き，特殊文字を削除している．文章の前処理の例を**表 2** に示す．

なお，ユーザ名が畳み込みの際の特徴の 1 つとならないよう，ユーザ名を He や She などの代名詞へ変換することも考えたが，何がユーザ名か判断することは容易ではな

表 1 サンプルのラベル付けの例
Table 1 Examples of labeling of samples.

ラベル	対象となる文章	読み手の反応
0	Got Pinterest again. Still not sure how to work it but will give it a go	you can also pin your blog posts and it might get more traffic!!
1	you never lie to me HAHAHAHAAHHA	It ended with you are an a **

© 2018 Information Processing Society of Japan

表 2 文章の前処理の例
Table 2 Examples of preprocessing of sentences.

処理前	Congrats to our CEO, @XXX on winning the ambient/instrumental category @XXX https://example... #yyy
処理後	Congrats to our CEO, @XXX on winning the category @XXX

表 3 サンプルの単語数
Table 3 Number of words of samples.

サンプル数：600
最大単語数：209
平均単語数：43.95333333333333
単語数の中央値：33.0
単語の母集団の分散：1466.9611555555557
標準偏差：38.33288630549151
100 単語以上のサンプルは，7.333333333333333%
200 単語以上のサンプルは，3.166666666666667%

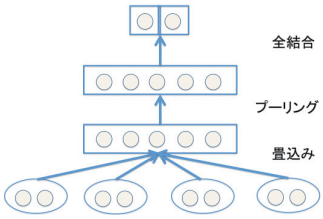

図 2 本研究の CNN モデルの構造
Fig. 2 Construction of CNN model of our research.

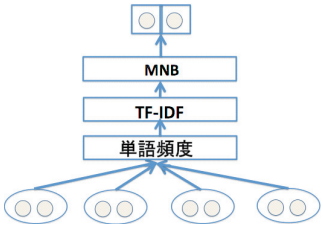

図 3 本研究の MNB 処理過程
Fig. 3 Procedure of process by MNB of our research.

かったため，本研究ではその変換は行っていない．この処理の後，1 サンプルの単語が 200 以上になった場合には，1 サンプルの大きさを揃えるため，新しく入力された単語から遡って 200 語までを抽出している．これは，CNN を使用する際に，サンプルのデータサイズを揃える必要があるためである．また，記述に改行が含まれている場合，この改行も重要な要素の 1 つと考え，<NL> へと変換している．

本研究において使用したサンプルの単語数を python の statistics という module を利用して調査したところ，**表 3** の結果となった．

表 3 より，100 単語以上のサンプルもかなり少ないが，最大の単語数が 209 という点を考慮して，ほとんどのサンプルの会話をすべて取り入れるために，本研究においては単語の最大長を 200 とした．なお，これを 209 にすればすべてのツイートのすべての内容が網羅できるし，201 や 199 が不適切である根拠はないが，単純に人間にとってきりの良い数字を選択した．この選択により，本研究においては，約 97%のツイートに関しては会話のすべてを分析に使用でき，残りのツイートに関しても，最大で末尾の 9 単語（200–9）が切り捨てられるのみとなる．もちろん，単語数を最大限多くしたほうが精度は高くなるが，分類における計算量も増大する．

3.5 CNN と MNB の 2 段フィルタによる分類

Twitter から収集された文章は，ラベル付けと前処理の後，ベクトルに変換される．本研究では，word2vec を用いて実装を行っている．word2vec では，Godin らが作成した Twitter のデータを用いた word2vec モデル [12] を用いた．

ベクトルに変換されたサンプルは，CNN と MNB の 2 つのフィルタをこの順で通過する．実装上は，2 つのフィルタを通過したサンプルの和集合をとっている．CNN と MNB のフィルタに使用されるモデルの構造と処理内容を本節で詳述する．

3.5.1 CNN フィルタのモデル詳細

CNN のフィルタに関しては，Y. Kim のモデル [9] に似たシンプルなネットワークを提案する．本研究で用いるモデルの構造を**図 2** に示す．なお，図中の丸は各層のノードを示し，最下のものが入力されたサンプル，最上のものが出力された分類結果を指す．

本研究では，畳み込み層の後にプーリング層を通した後，全結合層を通し，Softmax 関数を用いて兆候があるかどうかの 2 値に分類している．Y. Kim の研究と同じく，初めの畳み込み層において単語どうしの特徴を抽出し，プーリング層と全結合層において文章の意味を理解するという構造となっている．畳み込み層では，入力のチャネルは 1 次元とし，フィルタのサイズは 3×3 としている．そのため，畳み込み層ではフィルタがとらえるのは連続した 3 単語の特徴である．また，パディングは 1 であり，ストライドも 1 としている．その後，活性化関数として Relu 関数を出力に対して適用している．次に，プーリング層では，2×2 のフィルタから最大の値を持ってくる最大プーリングを行っている．その後の全結合層でも畳み込み層と同様に活性化関数は Relu 関数を適用している．なお，サンプルを学習させる際のエポック数は 100 とした．

3.5.2 MNB フィルタの処理詳細

MNB は自然言語処理などの離散的なデータに対してよく利用されるモデルである．この MNB を使ってサンプルを処理する過程を**図 3** に示す．なお，図中の丸はノードを示し，最下のものが入力されたサンプル，最上のものが出力された分類結果を指す．

MNB は負の値を受け取った学習ができないため，word2vec のようなツールを用いて単語を変換すること

© 2018 Information Processing Society of Japan

ができない．そこで，出現単語の頻度をもとにした単語ごとの TF-IDF の値を用いたベクトルを MNB に学習，評価させることとする．

4. 実装

本研究における，人が怒っているかどうかを判別するためのデータの収集に関する実装，また，そのデータを用いて学習，テストを行うフィルタについての実装について本章で詳述する．

4.1 データの取得

本実装のデータの取得には Twitter API を用いている．3.2 節で述べたデータを，2 段階に分けて取得する．1 段階目では，ハッシュタグをもとに，人が怒っている投稿を取得する．このとき，使用するハッシュタグは #angry，#fuck，#bitch の 3 つであり，そのハッシュタグを持つ投稿が他の投稿に対して返信してあり，かつ，英語で記述された投稿を取得している．

2 段階目では，ハッシュタグを用いて収集した，人が怒っている投稿をもとにして，返信がなくなるまで投稿を遡って取得し続ける．返信がなくなるまで投稿を取得し続けることによって，会話の始まりから人が怒るまでの一連の文章を取得している．これらのプログラムを，Python を用いて記述している．

なお，本実装におけるプログラムを動かす際，Twitter API の利用制限の関係上，15 分あたりに取得できるデータ量に上限がある．また，API の利用制限において，これらのデータは過去 7 日間分しか遡れない仕組みになっている．そのため，本手法でデータを大量に収集するためには，継続的にデータを収集しなければならない．

4.2 データのラベル付けと前処理

3.3 節および 3.4 節で示したように，データを収集した後に，データに対してラベル付けと前処理を行う．ラベル付けの対象となるデータは，4.1 節で述べた，怒りの兆候がある一連の投稿および，同様に Twitter API から得られた怒りの兆候がない英文の一連の投稿である．一連の投稿を 1 サンプルとし，怒りの兆候があるサンプルには “1” を，ないサンプルには “0” をラベル付けしている．また，収集された一連の投稿に関しては，Python の標準ライブラリに含まれている正規表現のライブラリを用いて，ハッシュタグと URL を取り除く前処理を行っている．

4.3 分類器の学習方法

分類器の学習にあたり，CNN とそれ以外の分類器に分けて，3.5 節で述べた手法に対して実装を行った．

4.3.1 CNN を用いた学習

3.5.1 項で述べた CNN を用いた学習のためのプログラムを，Python の chainer ライブラリを用いて実装した．なお，chainer の中で行列を用いるため，numpy という行列計算を行うためのライブラリも使用している．

4.2 節の前処理をしたサンプルを，学習用の訓練データと評価用のテストデータに分割し，word2vec を用いて単語をベクトル化する．なお，訓練データの中に word2vec のモデルに存在していない単語が含まれていた場合，ランダムなベクトルが生成されるようにしている．これは，モデルにない単語が特定のベクトルに変換されないようにすることで，複数回の学習の中でその単語が与える影響を少なくするためである．その後，chainer を用いて畳み込みニューラルネットワークのモデルを作成する．本実装では，畳み込み層を作成するために convolution 2d という関数を，プーリング層を作成するために pooling 2d という関数を用いている．もちろん，それ以外の層に関してもニューラルネットワークを作成するための関数は用意されている．作成したモデルに対して訓練データを学習させ，テストデータを用いて評価を行う．

4.3.2 MNB を用いた学習

3.5.2 項で述べた MNB のモデルを作成するにあたり，scikit-learn という機械学習のライブラリを用いて実装を行った．具体的には，CountVectorizer という関数を用いて，サンプルから単語の頻度を算出した後，TFIDFTransfer という関数を用いて TFIDF の値に変換したものを入力として MNB で学習を行う．

4.3.3 2 段フィルタの作成

3.5 節で述べた 2 段フィルタを作成する際，同じ訓練データを CNN と MNB のそれぞれに学習させる．その後，CNN と MNB それぞれのモデルにおいて，テストデータから間違えたサンプルを出力させる．2 段フィルタの外観としては，CNN でとりこぼしたサンプルを MNB で拾うということになっているが，実装上は CNN と MNB の独立したフィルタの出力の論理和となっている．本論文の評価においては，テストデータのサンプル数がそれほど多くないため，全サンプルを CNN と MNB それぞれのフィルタに通しても問題はないが，実運用する際にはサンプル数が膨大となるため，どちらか一方のフィルタでとりこぼしたサンプルのみをもう一方のフィルタに通すほうが効率が良い．

5. 評価

5.1 分類法による性能の評価

まず，本研究において使用する CNN や MNB を，我々が用意したデータの分類に適用すると，どの程度の精度で分類できるか評価を行った．本論文で用いている “精度” という用語であるが，正確に表現すると，機械学習においては Accuracy，Precision，Recall，F 値という 4 つの指標がある．Accuracy とは，テスト用の全サンプルのうち，正解したサンプルの割合を表す指標である．Precision とは，

© 2018 Information Processing Society of Japan

表 4 分類法ごとの性能比較
Table 4 Comparison of performance by methods.

分類法	Accuracy	Precision	Recall	F 値
CNN	0.78	0.81	0.79	0.79
MNB	0.76	0.77	0.77	0.76
SVM	0.65	0.66	0.66	0.66
CNN+MNB	0.92	0.91	0.92	0.92

表 5 分類法ごとの誤分類されたサンプル数
Table 5 Number of misclassification by methods.

分類器	兆候あり	兆候なし
CNN	5	14
MNB	13	8
SVM	15	16
CNN+MNB	2	5
CNN+SVM	3	6
MNB+SVM	5	5

出力として予測した対象のラベル（1 種類）のうち，正解したサンプルの割合を表す指標である．Recall とは，テスト用の全サンプルに対して，実際に存在している対象のラベル（1 種類）のうち，正確に予測できたサンプルの割合を表す指標である．F 値とは，Precision と Recall の調和平均であり，この値が高いほど使用した分類器の性能が高いとされている．

本評価においては，全サンプル 600 個が，訓練用の 510 個とテスト用の 90 個に分けられている．これは，訓練用に 85%，テスト用に 15%のサンプルを割り当てたためである．

評価に使用する分類法としては，本研究において使用する CNN や MNB を単独で用いるほか，感情分析を行う際によく用いられる分類法 [2], [3], [5], [6] である SVM も加えた．SVM におけるベクトル変換には word2vec を用いている．さらに，提案手法である，3.5 節で述べた 2 段フィルタについても評価を行った．これらの評価結果を**表 4** に示す．

表 4 より，感情分析を行う際によく用いられる SVM は他の分類法より少し精度が低く，提案手法の CNN と MNB による 2 段フィルタが最も精度が高かった．

5.2 誤分類についての評価

単純に精度の評価のみ行うと，人を怒らせるサンプル（兆候あり）とそうでない（兆候なし）がそれぞれどれくらい誤分類されたかが不明となる．兆候ありのサンプルに偏って誤分類される場合には，精度として示された値よりも人を怒らせる発言を見逃す確率が高くなり，兆候なしのサンプルに偏る場合には，人を怒らせない発言も投稿時に指摘される確率が高くなる．そこで，表 4 の評価において誤分類されたサンプル数を**表 5** に示した．なお，テスト用の 90 サンプルのうち，兆候ありは 30 個，なしは 60 個であった．表中には提案手法の 2 段フィルタである CNN+MNB

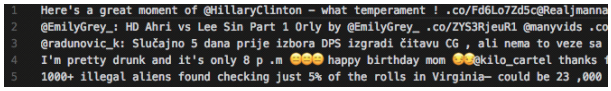

図 4 CNN によって誤分類された人を怒らせるサンプル
Fig. 4 Sample with making irritation by CNN misclassification.

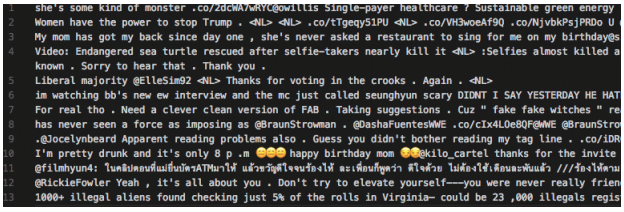

図 5 MNB によって誤分類された人を怒らせるサンプル
Fig. 5 Sample with making irritation by MNB misclassification.

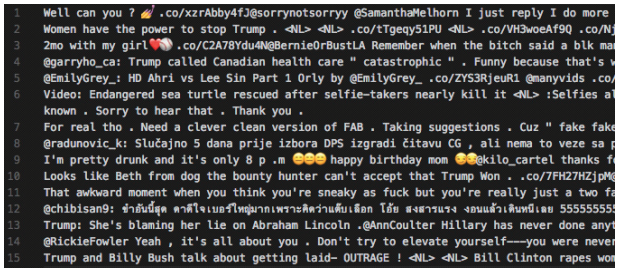

図 6 SVM によって誤分類された人を怒らせるサンプル
Fig. 6 Sample with making irritation by SVM misclassification.

のほかに，CNN+SVM および MNB+SVM の 2 段フィルタの評価も示しているが，これはなぜ我々の 2 段フィルタが CNN+MNB であるのか 6.2 節で考察するためである．

5.3 誤分類された人を怒らせるサンプル

誤分類された人を怒らせるサンプルにどのようなものが含まれているか調査した．これは，人間であれば絶対に見逃さないようなサンプルが誤分類されているか，また，人間でも判定が難しいようなサンプルが誤分類されているかを，各分類法において確認するためである．また，分類法ごとにどのようなサンプルを見逃しやすいのかについても調査した．CNN による例を**図 4** に，MNB による例を**図 5** に，SVM による例を**図 6** にそれぞれ示す．それぞれの図において，誤分類されたサンプルが 1 行ごとに出力されている．

提案手法の 2 段フィルタによる分類では，フィルタで検知されるものは CNN と MNB の和集合となるため，本 2 段フィルタで誤分類されるものは，図 4 と図 5 の双方に出現するサンプルとなる．ここでは，“I'm pretty . . .” で始まる文と “1000+ illegal . . .” で始まる 2 つの文が該当しており，表 4 でも兆候ありの誤分類は 2 つであった．

次に，ある分類法では正しく検知できるのに，他の分類

© 2018 Information Processing Society of Japan

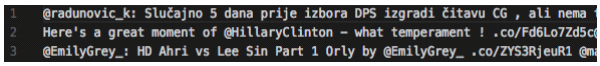

図 7 MNB で正しく分類できて CNN で見落とした人を怒らせるサンプル

Fig. 7 Sample with making irritation by MNB correct classification and CNN misclassification.

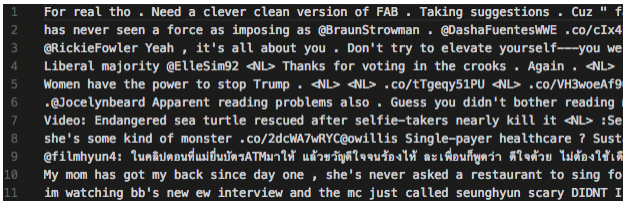

図 8 CNN で正しく分類できて MNB で見落とした人を怒らせるサンプル

Fig. 8 Sample with making irritation by CNN correct classification and MNB misclassification.

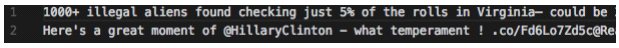

図 9 SVM で正しく分類できて CNN で見落とした人を怒らせるサンプル

Fig. 9 Sample with making irritation by SVM correct classification and CNN misclassification.

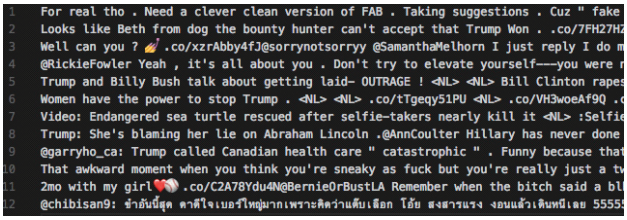

図 10 CNN で正しく分類できて SVM で見落とした人を怒らせるサンプル

Fig. 10 Sample with making irritation by CNN correct classification and SVM misclassification.

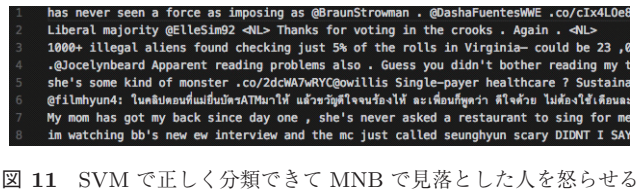

図 11 SVM で正しく分類できて MNB で見落とした人を怒らせるサンプル

Fig. 11 Sample with making irritation by SVM correct classification and MNB misclassification.

法では見落としてしまう人を怒らせるサンプルにはどのようなものがあるかについて調査した．MNB で正しく分類できて CNN で見落としたものを**図 7** に，CNN で正しく分類できて MNB で見落としたものを**図 8** に，SVM で正しく分類できて CNN で見落としたものを**図 9** に，CNN で正しく分類できて SVM で見落としたものを**図 10** に，SVM で正しく分類できて MNB で見落としたものを**図 11** に，MNB で正しく分類できて SVM で見落としたものを

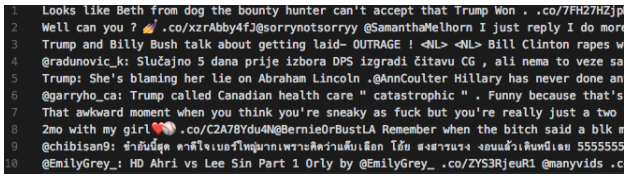

図 12 MNB で正しく分類できて SVM で見落とした人を怒らせるサンプル

Fig. 12 Sample with making irritation by MNB correct classification and SVM misclassification,

図 12 にそれぞれ示した．個々のサンプルに含まれる記述の詳細については，6.2 節で改めて示す．

6. 考察

表 4 より，人を怒らせる発言を検出する用途にそれぞれの分類法を用いた場合，F 値が CNN で 0.79，MNB で 0.76，SVM が 0.66 であった．いずれの分類法でも Precision と Recall のいずれかに数値が偏ることもなく，安定した学習が行えているといえる．感情分析を行う際によく用いられる分類法である SVM は他の分類法より F 値が 0.1 以上低く，今回のように，分析対象の記述が持つ感情ではなく，分析対象の記述によって引き起こされる他者の感情を分析するのには不向きであることが分かった．

一方で注目すべきは，本提案手法である 2 段フィルタを用いると，表 4 の CNN+MNB の F 値が 0.92 となっているように，分類精度が飛躍的に向上する点である．これはいい換えれば，CNN と MNB でとりこぼしてしまう記述にはそれぞれ何らかの特徴があり，両分類法が補い合える記述が一定程度存在するということである．

それぞれの分類法において，とりこぼしてしまうサンプルにどのような特性があるのか考察する．

6.1 CNN がとりこぼす人を怒らせる記述

CNN がとりこぼす人を怒らせる記述の例を**図 13**，**図 14**，**図 15**，**図 16**，**図 17** に示す．

図 13 には Hillary Clinton という人名が登場しており，政治に関する記述であると推測される．誤分類の理由としては，収集されたデータの中に政治に関する記述が少なかった点と，話題が大統領選挙の時期のものであり，特定の時期にしか話題にならない点が考えられる．CNN においては類似のサンプルが多く集まるほど精度が上がる傾向があり，より多くのデータを集められれば，このような記述も誤分類されなくなると思われる．

図 14 には “@” から始まるユーザ名が多く，人間にも判断が困難であると思われる．単語をベクトルに変換する際，このようなユーザ名もそれぞれ異なったベクトルにランダムで決まってしまうため，誤分類につながったと推測される．ただし，CNN の場合，このようなベクトルが一定量存在しても，人を怒らせる兆候となるベクトル群をうまく

© 2018 Information Processing Society of Japan

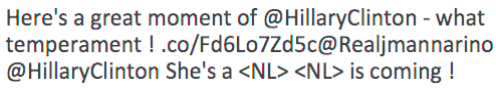

図 13 CNN によって誤分類された人を怒らせる記述例 1
Fig. 13 Example of description with making irritation by CNN misclassification 1.

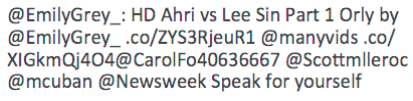

図 14 CNN によって誤分類された人を怒らせる記述例 2
Fig. 14 Example of description with making irritation by CNN misclassification 2.

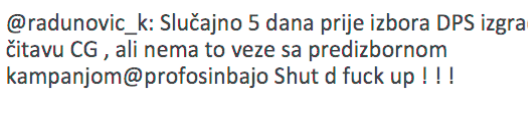

図 15 CNN によって誤分類された人を怒らせる記述例 3
Fig. 15 Example of description with making irritation by CNN misclassification 3.

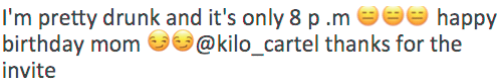

図 16 CNN によって誤分類された人を怒らせる記述例 4
Fig. 16 Example of description with making irritation by CNN misclassification 4.

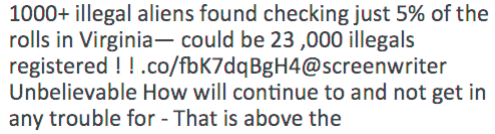

図 17 CNN によって誤分類された人を怒らせる記述例 5
Fig. 17 Example of description with making irritation by CNN misclassification 5.

畳み込めれば正しく分類できるため，MNB などの統計的な分類法では見分けられないものも見分けられる可能性はある．なお，すべてのユーザ名を同一ベクトルに変換してしまう方法もあるが，それでは図 13 の Hillary Clinton のような特定の人物名が特徴を失ってしまう．現時点では調査を行っていないが，登場頻度が高い特定の人物名に特別なベクトルを付与すればうまく分類できる可能性はある．

図 15 には，文中に英語以外の記述が含まれている．これは，Twitter を用いてデータを収集する際，Twitter API の仕様から，Tweet の属性が英語であるという投稿を集めたためである．本提案手法においては，言語のベクトル処理を行う word2vec のモデルを作成する段階で，英語のコーパスを学習させている．著者らが英語と日本語以外の言語は理解できないため，英語以外の言語は試していないが，他言語のコーパスも収集し，混在して学習させれば，図 15 のサンプルに示される記述も正しく分類できるのではないかと思われる．なお，日本語，中国語，韓国語のように，単語どうしの境界がない言語の場合には，本提案手法で述べた処理に加え，文章を単語に分割する形態素解析のような処理が必要となる．

図 16 に示される記述は，絵文字と文面から皮肉であることが分かる．これは，人間には容易に正しく分類できるが，ネガティブな文面ではないため，CNN が誤分類してしまう典型例の 1 つである．現時点では難しいが，皮肉の例を数多く学習させることができれば CNN でも正しく分類できると考えられる．

図 17 では，1000+ illegal aliens という記述において，移民（immigrant）を alian という隠語で罵倒している．これも，人間には正しく分類できるが，隠語を理解しない CNN が誤分類してしまう典型例の 1 つである．このような例も我々の手法では対応は困難であるが，隠語によって人を怒らせる例を数多く学習させることができれば CNN でも正しく分類できると考えられる．

以上より，本提案手法の CNN のフィルタにおいては，政治的発言，皮肉，隠語にうまく対応できないことが分かった．ただし，学習に使用するデータ量を増やせば精度を向上させられる可能性はある．なお，提案手法の CNN では畳み込みを 1 回しか行っていないが，畳み込み層を増やすことでも精度が上がるかもしれない．しかし，これは計算量の増加とトレードオフの関係にあるため，データ量が増えた場合に現実的な解であるかどうかは現時点では分からない．

また，英語以外の言語による記述や大量のユーザ名を含んだ記述も正しく分類できなかったが，これらは他の前処理を加えて排除すれば対応可能である．

6.2 CNN と SVM がとりこぼす人を怒らせる記述の差異

本節では，感情分析を行う際によく用いられる分類法である SVM と，提案手法が使用する CNN において，一方で正しく分類できてもう一方でとりこぼす人を怒らせる記述には，どのような差異があるのかについて考察する．SVM で誤分類され CNN で正しく分類された人を怒らせる記述例を**図 18**，**図 19**，**図 20**，**図 21** に示す．

図 18 に示される発言には，Monkey faces look better on monkeys という挑発的な記述がある．おそらく，CNN では挑発的な記述を畳み込みによりうまく見つけることができたが，monkey という単語自体そのものが他人を罵倒するものではないため，SVM では正しく分類できなかったのではないかと推測される．なお，この発言には Trump という単語が含まれており，政治的な内容も含まれている．

図 19 に示される発言には，Well can you?というやや挑発的な記述と，your bitch-ass mom という明らかに母親を侮辱する記述が見られる．しかし，発言全体を通して負の意味を持つ単語は bitch-ass しかないため，SVM では正しく分類されなかったのではないかと推測される．一方で

© 2018 Information Processing Society of Japan

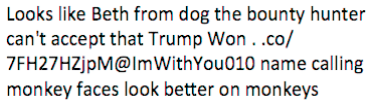

図 18 SVM で誤分類され CNN で正しく分類された人を怒らせる記述例 1

Fig. 18 Example of description with making irritation by SVM misclassification and CNN correct classification 1.

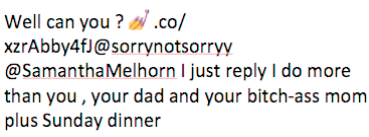

図 19 SVM で誤分類され CNN で正しく分類された人を怒らせる記述例 2

Fig. 19 Example of description with making irritation by SVM misclassification and CNN correct classification 2.

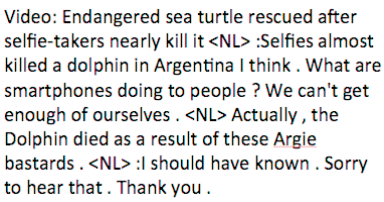

図 20 SVM で誤分類され CNN で正しく分類された人を怒らせる記述例 3

Fig. 20 Example of description with making irritation by SVM misclassification and CNN correct classification 3.

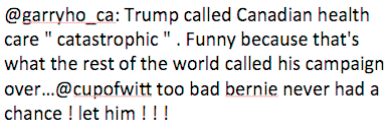

図 21 SVM で誤分類され CNN で正しく分類された人を怒らせる記述例 4

Fig. 21 Example of description with making irritation by SVM misclassification and CNN correct classification 4.

CNN では，前述の挑発的な記述や侮辱の記述を，ベクトル化された一連の単語群としてうまく畳み込めている．

図 20 の発言は，ウミガメやイルカが自撮りをする人々に殺されそうになったことに対して抗議をするものである．この発言には，kill や die などの負の意味を持つ単語が含まれているが，前述の bitch-ass などとは異なり，人を怒らせるかどうかに関しては強い特徴を持っていない．よって，これも図 19 の発言と同様，単語単位のベクトルで評価する SVM ではうまく分類できず，ベクトル化された一連の単語群を畳み込む CNN ではうまく分類できていると推測される．

図 21 も同様に，単語単位で見ると負の意味を持つ単語は catastrophic と bad くらいであり，これが SVM でうまく分類できなかった原因と推測される．また，Trump や health care，bernie という言葉から，これは政治的な発言である．おそらく，CNN では，Funny because や too bad bernie といった表現をうまく畳み込めたのではないかと思われる．

以上より，単語単位で評価を行う SVM と比較して，連続した単語を畳み込める CNN は，文意を解釈する必要があるような発言に対しては有利だといえる．

一方で，SVM では正しく分類されて CNN では誤分類された記述についても評価する．これらは，すでに図 13 と図 17 に示されている．SVM がこれらをうまく分類できた理由は定かではないが，おそらく特定の政治的な内容に現れる単語や隠語に独特のベクトルが含まれる場合があり，それがうまく見いだされたのではないかと推測される．

これらの考察から，SVM のフィルタは CNN のフィルタにほぼ包含されると我々は考える．その理由として，表 5 から，双方のフィルタが共通して誤分類した兆候ありのサンプルは 3 個しかないにもかかわらず，CNN がとりこぼして SVM が正しく分類できたサンプルは 2 個であり，逆に，SVM がとりこぼして CNN が正しく分類できたサンプルは 12 個も存在しているからである．これは，CNN が正しく分類できるサンプルの 40%を SVM は正しく分類できないことを示している．しかし，その一方で，CNN が誤分類した政治的内容や隠語を含んだサンプルを SVM は正しく分類できる可能性についても，本節で考察されている．ただし，これらは 6.1 節で述べたように，サンプルに使用できるデータ量さえ増加すれば，CNN でも正しく分類される可能性がある．

6.3 CNN と MNB がとりこぼす人を怒らせる記述の差異

本節では，提案手法の 2 段フィルタに使用する CNN と MNB において，一方で正しく分類できてもう一方でとりこぼす人を怒らせる記述には，どのような差異があるのかについて考察する．表 5 より，兆候ありに関して CNN の誤分類が 5 個，MNB の誤分類が 13 個，CNN+MNB の誤分類が 2 個であることから，CNN が正しく分類して MNB が誤分類したものは 11 個，その逆は 3 個となる．

MNB が正しく分類して CNN が誤分類したものは図 13，図 14，図 15 に示される 3 つの発言である．このように，CNN ではうまく畳み込めない記述であっても MNB では正しく分類できるものもある．

なお，MNB のフィルタも人を怒らせる記述に関しては，SVM のフィルタと同様に CNN のフィルタにほぼ包含されると我々は考える．その理由として，表 5 から，双方のフィルタが共通して誤分類した兆候ありのサンプルは 2 個しかないにもかかわらず，CNN がとりこぼして MNB が正しく分類できたサンプルは 3 個であり，逆に，MNB がとりこぼして CNN が正しく分類できたサンプルは 11 個も存在しているからである．これは，CNN が正しく分類できるサンプルの約 37%を MNB は正しく分類できないこ

© 2018 Information Processing Society of Japan

とを示している．

CNN がとりこぼすサンプルを MNB が正しく分類できる理由であるが，単語をベクトルに変換する段階において，単語の頻度を参考にすることによって文章の特徴を抽出し，意味のありそうな単語のみをうまく抜き出すという特徴抽出方法を行っているからであるのではないかと考えられる．これにより，前処理に失敗したサンプルを正しく判別できていると思われ，この点では SVM よりも MNB のほうが優位なのではないかと考えた．

6.4 CNN と MNB がとりこぼす人を怒らせない記述の差異

本節では，人を怒らせない記述について，CNN と MNB とのフィルタの差異を考察する．表 5 より，兆候なしに関して CNN の誤分類が 14 個，MNB の誤分類が 8 個，CNN+MNB の誤分類が 5 個であることから，CNN が正しく分類して MNB が誤分類したものは 3 個，その逆は 9 個となる．

MNB が正しく分類して CNN が誤分類したものの中から，代表的な 3 つのサンプルを**図 22**，**図 23**，**図 24** に示す．

図 22 では，一見すると bombard my Twitter crushes などの物騒な記述が見られるが，これは大げさな冗談で birthday について話しているものと思われる．CNN では，この物騒な表現を畳み込んで人を怒らせるものと判断したが，MNB では，bombard，crush，peep などの負の意味を持つ単語がある一方で，birthday や best のような正の意味を持つ単語も含まれているため，人を怒らせないと判断したと推測される．

It's my birthday next Sun so I need my peeps to bombard my Twitter crushes so they tweet me @RogerMathewsNJ @bulldog_nj @GabrielMacht ❤ <NL> @LisaTrampolina @RogerMathewsNJ @bulldog_nj @GabrielMacht Bryan especially will wish you the Best .

図 22 CNN で誤分類され MNB で正しく分類された人を怒らせない記述例 1

Fig. 22 Example of description without making irritation by CNN misclassification and MNB correct classification 1.

Bon , vous avez voté aux 2/3 pour un animé <NL> Donc ce sera sur un manga . <NL> @MisterEnzor Counnard

図 23 CNN で誤分類され MNB で正しく分類された人を怒らせない記述例 2

Fig. 23 Example of description without making irritation by CNN misclassification and MNB correct classification 2.

Sane , Stones , Bravo , a right-back and a striker . Happy ? <NL> @Llandegren Think we'll have to be happy .

図 24 CNN で誤分類され MNB で正しく分類された人を怒らせない記述例 3

Fig. 24 Example of description without making irritation by CNN misclassification and MNB correct classification 3.

図 23 における CNN の誤分類の理由は，この記述が英語ではないためである．提案手法においては，単語をベクトルに変換する際，学習されていない単語はランダムなベクトルに変換されるため，CNN の畳み込みに失敗したと推測される．MNB が正しく分類できた理由は，図 15 の場合と同様に，ランダムなベクトルを与えられた単語であっても，それらの出現状況から正しく分類が行える場合があるのではないかと推測される．

図 24 に示される発言は，まさに CNN の弱点を表している．この発言には，Happy のように正の意味を持つ単語があり，人を怒らせる要素は見当たらない．CNN は，学習過程において分類精度が最善になる重みに近づくように計算していくが，明らかに誤分類と判断されるサンプルが含まれているかどうかに関係なく，全体の精度が最大になるように計算を行う．この発言は，まさにこれに該当するサンプルと推測される．一方，MNB ではこのような問題は起きないため，この発言は正しく分類されている．

以上より，人を怒らせない記述に関しては，CNN のフィルタは MNB のフィルタにほぼ包含されると我々は考える．その理由として，表 5 から，双方のフィルタが共通して誤分類した兆候なしのサンプルは 5 個しかないにもかかわらず，MNB がとりこぼして CNN が正しく分類できたサンプルは 3 個であり，逆に，CNN がとりこぼして MNB が正しく分類できたサンプルは 9 個も存在しているからである．これは，MNB が正しく分類できるサンプルの約 15%を CNN は正しく分類できないことを示している．

6.5 2 段フィルタの組合せについて

6.2 節～6.4 節で比較したように，分類法によって正しく分類できるものとできないものがあることが分かる．表 5 より，CNN，MNB，SVM を比較した結果，兆候ありにおいては CNN が誤分類が一番少なく，兆候なしにおいては MNB が誤分類が一番少なくなっている．つまり，CNN と MNB は互いに補完関係にあり，提案手法のように CNN と MNB を組み合わせたフィルタが，人を怒らせる記述をフィルタリングするのに適しているといえる．

一方で SVM は単体で評価すると，兆候ありの場合には MNB と同程度の誤分類を，兆候なしの場合には CNN と同程度の誤分類をしており，あまり性能が良くない．最後に，表 5 の CNN+MNB，CNN+SVM，MNB+SVM の値を見ても，誤分類が一番少ないのは CNN+MNB である．

© 2018 Information Processing Society of Japan

6.6 使用したハッシュタグの妥当性について

3.2 節で述べたように，本研究で対象としたハッシュタグが #angry，#fuck，#bitch のみである理由として，2016 年 10 月 7 日から 10 月 13 日の間のツイートを，Twitter の API である REST API を用いて収集した結果，#angry が付与されたものは 106 個，#fuck のものは 136 個，#bitch のものは 385 個であり，それ以外のものは 30〜40 個にとどまったからであるとしているが，それ以外のものは本研究で対象としないデータとして，その数を正確に記録せずに破棄してしまった．そこで，再度，2017 年 8 月 19 日から 8 月 25 日の 7 日間に，同 API でツイートを取得し数を調査した．その結果，#angry が付与されたものは 62 個，#fuck のものは 136 個，#bitch のものは 206 個であり，本研究の対象から外したハッシュタグは，#anger が 41 個，#holyshit が 32 個，#cunt が 175 個，#asshole が 365 個，#bullshit が 654 個となった．よって，ハッシュタグが付けられたツイートの数は，週によって相当な違いがあることが分かった．

7. まとめ

SNS の発達により，誰もが自由に意見を世界中に発信できるようになった．しかし，思わぬ発言が人を傷つけてしまうこともある．そこで，我々は CNN と MNB を組み合わせた 2 段フィルタを用いて，これから発言されようとしているものが，過去に他人を怒らせた発言と類似しているかどうか，投稿前に確認する手法を提案した．本手法を用いて過去に他人を怒らせた発言に関して評価を行った結果，学習に使用しない評価用サンプルを使用して，人を怒らせた発言と怒らせなかった発言を 92.2%の精度（Accuracy と F 値の双方）で分類できた．一方で，今回の評価においては，Twitter API の仕様上やむをえない理由から学習に用いたデータ量が少なく，皮肉や隠語が用いられる発言をとりこぼしてしまった．しかし，データを多く集められさえすれば，CNN の特性上，皮肉や隠語にも対応できる可能性はある．提案手法の最大の特徴は，従来の研究のように，その発言が持つ感情をコンピュータを用いて分析するのではなく，その発言が他者に及ぼす影響を，過去の分類結果から予測する点にある．本手法を用いれば，過去に他人を怒らせた発言と類似している発言がある場合に，投稿前に利用者に知らせることが可能となる．将来，提案手法を用いたしくみが実用化すれば，利用者は投稿前に発言内容を再確認したり，発言自体をとりやめたりできる．

参考文献

[1] 総務省：国民のための情報セキュリティサイト，総務省（オンライン），入手先 ⟨http://www.soumu.go.jp/main_sosiki/joho_tsusin/security/index.html⟩（参照 2017-03-23）．

[2] Giachanou, A. and Crestani, F.: Like It or Not: A Survey of Twitter Sentiment Analysis Methods, *ACM Computing Surveys*, Vol.49, No.2, pp.1–41 (online), DOI: http://doi.org/10.1145/2938640 (2016).

[3] Mohammad, S.M.: #Emotional Tweets, *Proc. Intl. Conf. Lexical and Computational Semantics* (**SEM*), pp.246–255 (2012).

[4] Ekman, P.: An Argument for Basic Emotions, *Proc. Intl. Conf. Cognition and Emotion*, Vol.6, No.3, pp.169–200 (1992).

[5] Roberts, K., Roach, M.A., Johnson, J., Guthrie, J. and Harabagiu, S.M.: EmpaTweet: Annotating and Detecting Emotions on Twitter, *Proc. Intl. Conf. Language Resources and Evaluation* (*LREC*), pp.3806–3813 (2012).

[6] Sintsova, V., Musat, C. and Pu, P.: Semi-Supervised Method for Multi-Category Emotion Recognition in Tweets, *Proc. Intl. Conf. Data Mining Workshop* (*ICDMW*), pp. 393–402 (online), DOI: http://doi.org/10.1109/ICDMW.2014.146 (2014).

[7] Plutchik, R.: *The Psychology and Biology of Emotion*, HarperCollins College Publishers (1994).

[8] Kim, S., Bak, J.Y. and Oh, A.: Do You Feel What I Feel? Social Aspects of Emotions in Twitter Conversations, *Proc. Intl. Conf. Weblogs and Social Media* (*ICWSM*), pp.495–498 (2012).

[9] Kim, Y.: Convolutional Neural Networks for Sentence Classification, *Proc. Intl. Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), pp.1746–1751 (2014).

[10] Kalchbrenner, N., Grefenstette, E. and Blunsom, P.: A Convolutional Neural Network for Modelling Sentences, *Proc. 52nd Annual Meeting of the Association for Computational Linguistics*, pp.655–665 (2014).

[11] Dini, L. and Bittar, A.: Emotion Analysis on Twitter: the Hidden Challenge, *Proc. Intl. Conf. Language Resources and Evaluation* (*LREC*), pp.3953–3958 (2016).

[12] Godin, F., Vandersmissen, B., Neve, W.D. and de Walle, R.V.: Multimedia Lab @ ACL W-NUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations, *Proc. ACL 2015 Workshop on Noisy User-generated Text*, pp.146–153 (2015).

村山 大騎

2017 年東京工科大学コンピュータサイエンス学部卒業．現在，ソフトバンク株式会社勤務．

© 2018 Information Processing Society of Japan

宇田 隆哉 （正会員）

1998 年慶應義塾大学理工学部計測工学科卒業．2000 年同大学大学院理工学研究科計測工学専攻前期博士課程修了．2002 年同大学院理工学研究科開放環境科学専攻後期博士課程修了．博士（工学）．現在，東京工科大学コンピュータサイエンス学部講師．ネットワークセキュリティの研究に従事．2002 年 IFIP/SEC 2002 Best Student Paper Award 受賞．電子情報通信学会会員．

© 2018 Information Processing Society of Japan