インターネット上における短文投稿の攻撃性評価

古林　佑海　　（指導教員　西村　俊二）

平成31年1月25日

Evaluation of aggressiveness of short sentences on the Internet

Yumi Kobayashi　　(ACADEMIC ADVISOR　Shunji Nishimura)

概要：近年，スマートフォンの普及に伴い，SNSやTwitterなども普及し，ネットいじめなどが問題視されている．そのような問題を避けるために，様々な対策が取られてきた．単語単位による検出等の従来の手法では，誤った判定をされることが多く，その場合，それらの対策自体が無意味となる．ユーザーの報告によるユーザー規制では，悪意のあるユーザーにより，攻撃的な投稿をしていないユーザーが規制を受ける場合がある．より正確に，誹謗中傷や攻撃的な発言を判別するには，他の方法を採る必要があると考えられる．本研究では，Twitterより収集したランダムなツイートをもとに，自然言語処理ライブラリであるfastTextを用いて，単語の分散表現を獲得し，その分散表現をもとに，Twitterに投稿された短文の分散表現を獲得する．同様の手順で，Twitterに投稿された攻撃的な短文の分散表現を獲得し，Kmeans法によりその攻撃的な短文のクラスタリングを行う．最後に，短文の攻撃性に関するアンケートを，ランダムなツイートを用いて行い，同様のツイートの分散表現と，クラスタリングによって得た攻撃的な文章の分散表現とのコサイン類似度と比較を行い，提案手法の精度を判定する．提案手法の指標とアンケートの指標に関して，スピアマンの順位相関係数を用いた無相関検定を行い，有意水準1%において，これら2つの指標には相関があるという結果を得た．

キーワード: Twitter，fastText，MeCab

1. 緒言

1.1 背景

現在，SNS や Twitter は若い年代において必要不可欠なコミュニケーションツールとなったが，誹謗中傷やネットいじめで不快な思いをしたことのある人は多数存在する．これまで，単語単位の一致によるネット上への書き込みの規制や，ユーザーからの報告をもとに，ユーザー自体の規制をかける手法がとられてきた．単語単位の検出では，意図しない部分での単語の成立により，攻撃的な文章ではない場合でも規制の対象となる場合がある．例として，「自殺するのは良くない」という文章では，「殺す」という単語が検出される可能性がある．また，ユーザーの報告により規制を行う手法では，悪意のあるユーザーによって，攻撃的な書き込みをしていないユーザーが，不当な規制を受ける場合がある．このような誤検出，誤判定がされないような方法が求められている．

1.2 目的

本研究の目的は，単語単位の検出等の従来の手法を用いずに，インターネット上に投稿された短文の攻撃性を評価する手法を提案することである．

本稿では，文章を分散表現で表し，その分散表現を用いて，文章間の類似度を推測し，文章の攻撃性を評価する手法を提案する．より具体的には，まず多数のアクティブなユーザーのアカウントより投稿されたツイートを複数件収集し，それらのデータを整形した後，それらのデータを用いて，文章データを学習することにより単語の分散表現を獲得することのできる fastText[1]のモデルを作成する．その後，そのモデルを利用し攻撃的であると思われるツイートの分散表現を，そのツイートに含まれる単語の分散表現から獲得する．これにより，攻撃的であるかそうではないかの判定基準を作成する．この手法の精度評価はアンケートとの比較によって行う．

これまでの研究では，fastText の前身であり単語の分散表現を獲得することのできる word2vec[2]を用いての性格分析や[3]，単語間の類似度により Twitter 上に投稿された文章の感情を推定する研究[4]，単語間の類似度により炎上表現の単語の訂正を行うシステムを構築する研究[5]等が行われてきた．また，fastText を用いた研究は，外部の意味辞書を用いて，同義語を考慮した日本語の分散表現の獲得が行われている[6]．

1.3 本論文の構成

本稿の構成は以下の通りである．2 章では提案手法で用いるライブラリ及びエンジンの概要について述べ，3 章では，提案手法の具体的な内容と，提案手法の評価方法について述べる．4 章では実験結果とし

て，文章の攻撃性に関するアンケートと提案手法の比較を行い，その結果について考察を行う．5 章では本研究の今後の課題について述べる．6 章では本論文の結論をまとめる．

2. 本研究で用いる技術

本章では fastText，MeCab の概要を述べる．2.1 では fastText の概要及び，単語の分散表現の獲得方法について説明する．2.2 では，MeCab の概要と動作について説明する．

2.1 fastText

fastText とは，Facebook 社が公開したオープンソース自然言語処理ライブラリである．単語で区切られた文章を学習することにより，単語の前後関係から，それぞれの単語の分散表現(ベクトル)を獲得することが可能である．

fastText のモデルには，周辺の単語から，それらの単語の中心の単語を推測する Continuous Bag-of-Word と，対象の単語から，その単語の周辺の単語を推測する skip-gram が使われている[1]．本節では，本研究で用いた skip-gram について説明する．このモデルは，各一つずつの入力層，隠れ層，出力層からなる図 1，図 2 のようなニューラルネットワークのモデルである．学習に用いる単語の総数を n，獲得する単語のベクトルを m 次元とする．まず，学習に用いるテキストデータから，リスト 1 のように，重複のない単語の集合 V を得る．その後，周辺の何単語を予測するかのパラメータに従い，リスト 2 のような対象の単語とその周辺の単語の組み合わせを，それぞれのモデルの入出力として，図 1，図 2 のように学習を行う．この時，入力層から隠れ層への重み行列 W が n×m となり，隠れ層から出力層への重み行列 W’ が m×n となる．入力層と出力層の値について，用いられる値は one-hot ベクトルである．one-hot ベクトルは，リスト 3 のように，学習に用いる全単語数と同じ要素数をもち，各要素がすべての単語と一対一で対応したベクトルである．該当する単語の要素のみ 1 とし，他の要素を 0 として，図 1，図 2 のように入力層および出力層で用いる．この入力層の値と出力層の値の組み合わせが，学習に用いるデータの文章から得た，対象の単語とその周辺の単語との組み合わせと同様になる確率が大きくなるように，学習を行う．リスト 4 のように，W が単語の特徴を表したベクトルとなる．

リスト 1　単語の集合の例

```
文章「吾輩 は 猫 で ある」
     「名前 は まだ ない」
```

```
V = {吾輩, は, 猫, で, ある, 名前, まだ, ない}
```

リスト 2　周辺ごとの組み合わせの例

```
(吾輩, は), (は, 猫) , (は, 吾輩) , (猫, で)…
```

リスト 3　one-hot ベクトルの例

```
吾輩 ： (1, 0, 0, 0, 0, 0, 0, 0)
は   ： (0, 1, 0, 0, 0, 0, 0, 0)
猫   ： (0, 0, 1, 0, 0, 0, 0, 0)
で   ： (0, 0, 0, 1, 0, 0, 0, 0)
ある ： (0, 0, 0, 0, 1, 0, 0, 0)  …
```

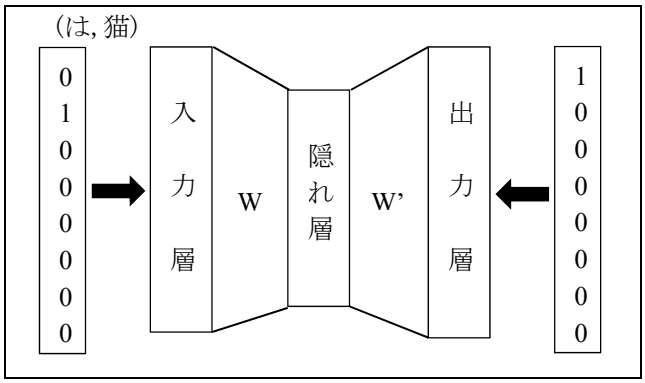

図 1　skip-gram の入出力の例 1

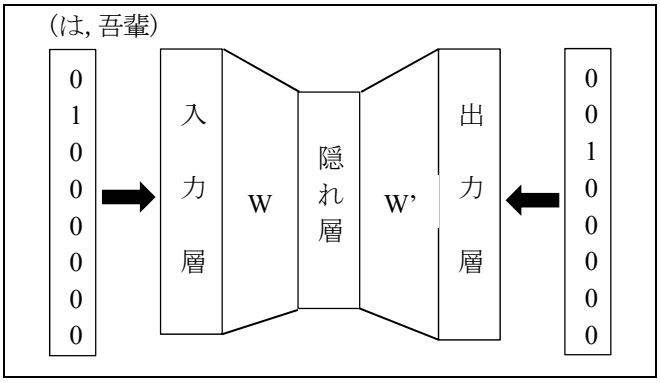

図 2 skip-gram の入出力の例 2

リスト 4　入力層から隠れ層への重み行列の例

$$W = \begin{pmatrix} 0.45\ldots & \cdots & 0.67\ldots \\ \vdots & \ddots & \vdots \\ 0.19\ldots & \cdots & 0.34\ldots \end{pmatrix} \begin{matrix} n \\ 行 \end{matrix}$$

m 列

2.2　MeCab

MeCab [7]とは，工藤拓によって開発されたオープンソース形態素解析エンジンである．リスト 5 のように，日本語の文章を，単語ごとに分かち書きすること

が可能である．また，リスト6のように，単語ごとに，品詞，活用形等の解析が可能である．

リスト5　分かち書きの例

```
昨日もテニスの練習をしました
昨日 も テニス の 練習 を し まし た
```

リスト6　解析の例

```
品詞,品詞細分類1,品詞細分類2,品詞細分類3,活用型,活用形,原形,読み,発音

すももももももものうち
すもも　　名詞,一般,*,*,*,*,すもも,スモモ,スモモ
も　　助詞,係助詞,*,*,*,*,も,モ,モ
もも　　名詞,一般,*,*,*,*,もも,モモ,モモ
も　　助詞,係助詞,*,*,*,*,も,モ,モ
もも　　名詞,一般,*,*,*,*,もも,モモ,モモ
の　　助詞,連体化,*,*,*,*,の,ノ,ノ
うち　　名詞,非自立,副詞可能,*,*,*,うち,ウチ,ウ
```

3. 提案手法と評価実験

本章では提案手法の概要及びその手順について説明する. 3.1 では学習データの収集及び収集したデータの整形の手法について述べ, 3.2 では文章ベクトルの獲得方法について述べる．3.3 では，文章の攻撃性を判定する際に用いる判定基準の作成方法について述べる．3.4 ではこの手法の精度評価の概要，及び精度評価に用いるアンケートの詳細について述べる．

3.1 学習データの収集および整形

3.1.1 ランダムなツイートの収集

TwitterAPI[8]を用いて，fastText の学習に用いるため，日本語のツイートの収集する．収集の対象としたアカウントは，日本の Twitter ユーザーであり，ユーザーがツイートを投稿するのではなく，自動でツイートが投稿されるアカウント(bot)ではないと判断できる，アクティブなユーザーのアカウントである．アクティブなユーザーのアカウントであるかどうかは，ツイートを収集する際に，直近数分でツイートが投稿されたかどうかで判断し，bot であるかどうかの判断は，ツイートを行うクライアントの名前から行う．それらのアカウントの直近のツイートを約 100 件ずつ収集する．収集を行う対象となるツイートからは，ハッシュタグ，URL，画像，アカウント名を除外する．その後，bot によるツイートが含まれているかを，100 件のツイートを目視することにより確認する．

3.1.2 収集したツイートの整形

fastText の学習に用いるために, 収集したツイートデータの整形を行う．2.1 で述べたように，学習に用いる文章は, 単語で区切られていなければならない. また，収集したツイートには，アクティブなユーザーのアカウントが投稿したツイートであるが，bot を用いて定期的に投稿される定期ツイートが含まれている可能性がある．

このように，収集した日本語のツイートをそのまま fastText の学習に用いるのは不可能であるため, 収集したツイートを整形する必要がある．手順は以下の通りである．

① 定期ツイートを削除するために，収集したツイートの中で重複する文章を削除する．
② 日本語文字列のコンバータである mojimoji[9]を用いて，学習の精度向上を目的に，テキストの全角，半角を統一する．
③ オープンソース 形態素解析エンジンである MeCab を用いて，文章の分かち書きを行う．インターネット上の新語や固有名詞に対応するために，MeCab のシステム辞書には，多数の Web 上の言語資源から得た新語を追加することでカスタマイズされた辞書である mecab-ipadic-NEologd[10]と，MeCab 標準のシステム辞書を併用する．

3.2 文章ベクトルの獲得

fastText を用いて 3.1 で作成したデータを学習させ，fastText のモデルを得る．モデルには，学習に用いた単語のベクトルが記憶されており，このモデルを用いることにより，単語のベクトルを求めることができる．学習時に存在していなかった単語についてはベクトルを求めることができない．

文章ベクトルの獲得の方法について，まず，入力された文章について，収集したツイートと同様に，全半角を統一し，MeCab を用いて分かち書きを行う．その後，分かち書きによって得た単語を一つずつベクトル化する．それらの単語のベクトルをすべて加算し，文章の単語数で除算することにより，相加平均を得る．この相加平均をその文章の文章ベクトルとする．

3.3 判定基準の作成

攻撃的と思われる文章を Twitter から約 500 件収集し, 3.2 で述べた手法により, それぞれの文章ベクトルを求める．その後，それらの文章ベクトルを SpectralEmbedding[11]により 3 次元に次元圧縮し，コサイン類似度に関する Kmeans 法[12]を用いて複数のクラスタにクラスタリングする．次元圧縮の手法として SpectralEmbedding を採用した理由について, 収集する攻撃的な文章のベクトルは，サンプル数が 1 万以下で非線形なデータとなるためである．クラスタリング

の手法として Kmeans 法を採用した理由について，収集する攻撃的な文章のベクトルは，サンプル数が 1 万以下であり，また，Kmeans 法は他の手法に比べ，コサイン類似度を距離関数とすることが容易であるためである．それぞれのクラスタに属している文章ベクトルを，クラスタごとにすべて加算した後に，そのクラスタに属している文章の総数で除算し，属している文章の文章ベクトルの相加平均を求め，そのベクトルをそのクラスタの代表の文章ベクトルとする．

3.4 アンケートとの比較

提案手法の精度の評価として，アンケートとの比較を行う．評価には Twitter から集めたランダムな 50 件のツイートを用いる．アンケートについて，対象のツイートが「攻撃的である」，「どちらかと言えば攻撃的である」，「どちらとも言えない」，「どちらかと言えば攻撃的ではない」，「攻撃的ではない」の 5 段階評価で実施する. 比較に用いるアンケート点は,「攻撃的である」から順に 1 点，0.75 点，0.5 点，0.25 点，0 点として，それぞれのツイートについて，その段階であると回答した点数をすべて加算した後，回答した人数で除算し，そのツイートのアンケートに関する相加平均を求め，アンケート点とする．提案手法の点数については，すべてのクラスタの判定基準の代表の文章ベクトルと，3.2 と同様の手順で求めた，対象のツイートの文章ベクトルとのコサイン類似度を求め，それらの中で最大値となった値を点数とする. これにより，それぞれの点が高ければ高いほど，攻撃的である文章と判断できる．アンケート点と判定基準との比較として, アンケートに用いたそれぞれのツイートについて，両者の点数の差異について考察する．また，アンケートの結果と提案手法により得た指標を散布図で図式化し，相関係数及び p 値を求め，相関関係について考察する．

4. 実験結果と考察

本章では，アンケートと提案手法の比較及び相関関係について述べる. 4.1 では本実験で収集したツイートについて述べる. 4.2 ではアンケートと提案手法の比較として，正しく判定がなされた例とそうでない例を挙げ，結果を考察する．4.3 では，アンケートと提案手法の相関関係の有無について述べる．

4.1 収集したツイートと fastText のモデル

3.1 について，本実験では合計で約 323 万件のツイートを収集した．また，目視で，100 件のツイートに bot と思われるツイートは含まれていないことを確認した. 3.2 について，データ全体で 5 回以上出現する単語を対象に，獲得する次元数は 150，学習の対象となる周辺の単語数は 5 とした．このモデルの総単語数は 14,532 個となった．また，3.3 について，目視で攻撃的であると思われるツイートを 364 件収集した．それらのツイートをクラスタリングした結果は以下の通りである．

表 1　クラスタリングの結果

番号	文章の例
1	うるせえよくず，黙れ帰れ無能，黙れちび，黙れころす，関西人黙れ👆，黙れ欠点野郎
2	死んだ?あれ?，タヒね，2 万回タヒね，油ってなんだよ，控えめに言ってタヒね
3	だまれよぶす，だまれよ下痢，オメェがだまれや，ねーよだまれごみ，だまれざこ
4	56 すぞ，あ?ぶち 56 すぞ?， 56 す，ぶち 56 すぞ, 56 すぞ w, 56 すぞ, おまえ 56 すよ???
5	陰キャ乙，だからデブやねんアスペ，主語をちゃんと書けアスペ，死ねクソメガネ
6	だまれお前がしね，だまれ豚土下座して謝れ，クソガキ，だまれゴリラお前がしね

4.2 アンケートと提案手法の比較

本実験では，10 代 8 名，20 代 5 名，それ以上の年代 1 名の計 14 名に対してアンケートを行った．

判定基準とのコサイン類似度が低い値を示していないが，アンケートの点数が低く，その文章が攻撃的ではないにもかかわらず，攻撃的であると誤判定する可能性がある文章の例，及びそのアンケートの点数と提案手法の点数は以下の通りである．

表 2　誤判定の可能性がある例 1

番号	文章
1	ねぇまってボロ泣きしてる
2	CDJ も電気出るし行きたかった
3	頼むぞ……
4	インフルエンザ A になったー
5	久々にカラオケ来た。
6	早く起きて午前に作業したいわ
7	どうすれば買ってくれると思う？

表 3　表 2 の文章のアンケート及び提案手法の点数

文章の番号	アンケート	提案手法
1	0.0714	0.5617
2	0.0714	0.5738
3	0.1071	0.6158
4	0.1250	0.6192
5	0.1250	0.5890
6	0.1428	0.6338

7	0.1607	0.6317

表 2，表 3 について，文章 4，文章 5，文章 7 が表 3 のような結果となったのは，記号による影響であると考えられる．3.3 で判定基準を作成した際，収集した攻撃的な文章の中に「ー」が 16 件，「。」が 15 件，「?」が 23 件含まれており，判定基準とアンケートに用いた文章の類似度を計算する際，それらを表す分散表現の値が影響したためであると考えられる．文章 1，文章 2，文章 3，文章 6，文章 7 が表 3 のような結果となったのは，文章の単語数及びその内容が影響していると考えられる．3.3 で判定基準を作成した際，収集した攻撃的な文章の中に，助詞の「て」が 15 件，同じく助詞の「ぞ」が 54 件，「し」に関する動詞と「し」を含む文章が 26 件含まれており，判定基準とアンケートに用いた文章の類似度を計算する際，それらを表す分散表現の値が影響したと考えられる．また，単語数が増えることにより，文章のベクトルを正しく獲得できなかったため，表 3 のような結果になったと考えられる．

判定基準とのコサイン類似度が高い値を示していないが，アンケートの点数が高く，攻撃的であるにもかかわらず，攻撃的ではないと誤判定する可能性がある文章の例，及びそのアンケートの点数と提案手法の点数は以下の通りである．

表 4　誤判定の可能性がある例 2

番号	文章
8	叩いたら殺す
9	死ね帰れ

表 5　表 4 の文章のアンケート及び提案手法の点数

文章の番号	アンケート	提案手法
8	0.9642	0.6928
9	1.0000	0.7028

表 4，表 5 の文章 8，文章 9 について，文章に含まれている単語の種類が影響を及ぼし，このような結果となったと考えられる．3.3 で判定基準を作成する際に収集したツイートの中に，「叩いたら」に含まれる動詞の「叩い」，またそれらに関係する単語を含む文章は存在せず，「帰れ」が含まれる文章は 2 件のみ含まれていた．このように，収集した攻撃的な文章の中にコサイン類似度の高い単語，もしくは同様の単語が多く含まれていなかったため，表 5 のような結果になったと考えられる．しかしながら，逆に「死ね」と「殺す」，助動詞の「たら」とのコサイン類似度が高い単語である「て」は，収集したツイートの中に，それぞれ 33 件，16 件，15 件と多数含まれていたため，文章に含まれる単語数が少ないにもかかわらず，判定基準とのコサイン類似度が高くなったとも考えられる．

判定基準とのコサイン類似度が低く，アンケート点も低くなり，正しく判定がなされたといえる文章，及びそのアンケートの点数と提案手法の点数は以下の通りである．

表 6　正しく判定がなされた例 1

番号	文章
10	ただいま新潟
11	いざ飛行機
12	さようなら静岡

表 7　表 6 の文章のアンケート及び提案手法の点数

文章の番号	アンケート	提案手法
10	0.0535	0.3241
11	0.1785	0.4184
12	0.0714	0.4568

表 6，表 7 について，正しく判定がなされた理由としては，文章に含まれている単語の種類，単語数があげられる．3.3 で収集したツイートの中に，文章 10，文章 11，文章 12 に含まれている単語と同様の単語，及びコサイン類似度の高い単語が含まれていなかったため，表 7 のような結果になったと考えられる．また，これらの文章に含まれている単語数が少なく，文章のベクトルを正しく獲得できたことも影響していると考えられる．

判定基準とのコサイン類似度が高く，アンケート点も高くなり，正しく判定がなされたといえる文章，及びそのアンケートの点数と提案手法の点数は以下の通りである．

表 8　正しく判定がなされた例 2

番号	文章
13	は？56 すよ？
14	うんこ黙れよ
15	控えめに言ってタヒね
16	黙れじじい
17	どっちにしろ雑魚乙
18	お前アスペかよ

表 9　表 8 の文章のアンケート及び提案手法の点数

文章の番号	アンケート	提案手法
13	0.9821	0.9659
14	1.0000	0.9108
15	0.9642	0.9261
16	1.0000	0.8732
17	0.9285	0.8138
18	0.9285	0.8161

表 8，表 9 について，正しく判定がなされた理由としては，文章に含まれている単語の種類があげられる．文章 13 から文章 17 について，3.3 で収集したツイートの中に，「?」が 23 件，「タヒ」が 25 件，「お前」が 19 件，「56」が 60 件，「雑魚」と「ザコ」が合わせて 21 件，「黙れ」が 38 件含まれており，また，それらとコサイン類似度の高い単語も多く含まれていたため，表 9 のような結果になったと考えられる．文章 18 について，3.3 で収集したツイートの中に「アスペ」は 8 件しか含まれていなかったものの，同じく 3.3 でクラスタリングした際に，表 1 のクラスタ 5 のように，「アスペ」を含む文章は同様のクラスタに属することとなり，そのクラスタの代表値とのコサイン類似度が高くなったため，結果として提案手法の点数が高くなったと考えられる．

判定基準とのコサイン類似度，及びアンケート点ともに中間の値を示し，正しく判定がなされたといえる文章，及びそのアンケートの点数と提案手法の点数は以下の通りである．

表 10　正しく判定がなされた例 3

番号	文章
19	サンボ聴いてる座ってる www
20	は?さむ
21	遠距離陰キャ戦法すれば勝てるゲームやからクソ
22	スマホぶん投げそうになったわ
23	なんで小文字にした？

表 11　表 10 のアンケート及び提案手法の点数

文章の番号	アンケート	提案手法
19	0.3571	0.5146
20	0.4464	0.5008
21	0.7500	0.7890
22	0.5535	0.6876
23	0.6071	0.6757

表 10，表 11 について，文章 20，文章 23 が表 11 のような結果となった理由として，表 8，表 9 の例と同様に，3.3 で収集したツイートの中に，「?」が 23 件と多数含まれていたためであると考えられる．文章 19，文章 22 が表 11 のような結果となった理由としては，3.3 で収集したツイートの中に，「w」が 7 件，「わ」が 5 件と少ないながらも含まれていたためであると考えられる．文章 21 について，3.3 で収集したツイートの中に「陰キャ」は 3 件，「クソ」は 5 件含まれており，多数含まれていたわけではない．しかしながら，同じく 3.3 でクラスタリングした際に，表 1 のクラスタ 5 のように，「陰キャ」，「クソ」及びそれらの単語とコサイン類似度が高い「糞」等を含む文章が同じクラスタに多数属していたため，表 11 のような結果になったと考えられる．

1 で述べた単語単位の一致による検出との比較となる文章及びそのアンケートの点数と提案手法の点数は以下の通りである．

表 12　単語単位の検出と比較となる文章

番号	文章
24	これ死ねる
25	殺す気?
26	死ね帰れ
27	叩いたら殺す

表 13　表 12 のアンケート及び提案手法の点数

文章の番号	アンケート	提案手法
24	0.2500	0.5524
25	0.5892	0.7335
26	0.9642	0.6928
27	1.0000	0.7028

表 12，表 13 について，文章 24，文章 25 は，提案手法，アンケートともに高い値を示さないが，単語単位による検出では規制となるような例であり，文章 26，文章 27 は，単語単位による検出では規制対象となるが，提案手法では高い値を示さない例である．文章 24，文章 25 について，アンケート点は低い値もしくは中間の値を示し，攻撃的ではないと判断されるが，単語単位による検出を用いて，「死ね」，「殺す」を規制の対象とした際に，これらの文章は規制の対象となる．提案手法では高い値を示しておらず，攻撃的であるとは判断されていない．これは，単語単位による検出では意図しない規制を受ける文章であっても，提案手法では規制を受ける可能性が低い例であるといえる．文章 26，文章 27 について，アンケート点は高い値をとるにもかかわらず，提案手法では高い値を示しておらず，攻撃的であると判断する人が多いが，攻撃的ではないと誤判定を行う可能性がある．単語単位の検出を用いて「死ね」，「殺す」を規制の対象とした際には，これらの文章は規制の対象となり，正しく判定がなされることとなる．これは，判定手法では正しい判定がなされないが，単語単位による検出では正しく規制を行う例であるといえる．

これらの結果はすべて，3.3 で収集したツイートに起因していると考えられる．3.2 で文章のベクトルを獲得する際に，攻撃的な文章と攻撃的ではない文章の両方に，共通して数多く含まれる単語や記号の単語ベクトルによって，攻撃的な文章と攻撃的ではない文章の類似度が高くなったと推測される．しかしながら，攻撃的な文章にのみ含まれる単語と，その単語とのコサイン類似度の高い単語によって，攻撃的な文章を検出できたとも考えられる．

4.3 アンケートと提案手法の相関係数

3.4 で得たアンケート点と判定基準との類似度についてのグラフは以下の通りである．

実験結果の相関関係について，アンケートの結果及び提案手法の点数(判定基準とのコサイン類似度)は正規分布に従っていないので，スピアマンの順位相関係数[13]により，アンケートの結果と提案手法の点数の間に相関関係があるかについての検定を行った．スピアマンの順位相関係数は 0.7423661，p 値は 6.808e−10 となった．p 値について 6.808e−10<0.01 となったため，有意水準 1%において，アンケートの結果と提案手法の点数に相関関係があるといえる．

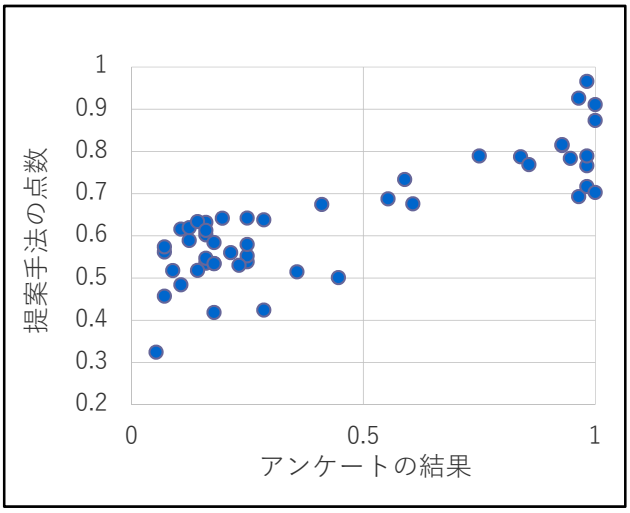

図 3　提案手法の点数とアンケートの点数の比較

5. 今後の課題

本研究の今後の課題について以下に述べる．

(1)fastText を用いて単語の分散表現を獲得する際，文書内における出現頻度である tf 値と，その単語が含まれている文章数及びその単語の逆文書頻度を表す idf 値を用いて単語の分散表現の重み付けを行い，単語の分散表現の精度向上を実現させる．

(2) Facebook 社が fastText の後に公開したオープンソース自然言語処理ライブラリである StarSpace[14]を用いて獲得した単語の分散表現を獲得する．その後，Sparse Composite Document Vectors[15]を用いて，それらの分散表現のクラスタリング，重み付けを行い，単語及び文章の類似性を考慮したうえで，単語の分散表現の精度向上を実現させる．

(3)「ーーー」や「っっっ」のように，同じ文字が重ねて現れるような文章を整形する．または「iphone」と「アイフォン」のような全く意味が同様である単語を，ひとまとめにして扱うなどの手順を試行し，3.1.2 で行ったデータの整形よりも，単語の分散表現を獲得するうえで有効なデータ整形の手順を求め，単語の分散表現の精度向上を図る．

(4) 日本語の意味辞書であり，対象の単語の同義語，上位語，下位語を獲得することの出来る日本語 WordNet[16]を用いて，単語の分散表現の精度向上を図る．

(5) SVM に基づく日本語係り受け解析器である Cabocha[17]を用いた係り受け解析により，単語間の関係から，何が攻撃の対象となっているのかを判断する手法を検討する

(6) 時系列データを扱うことのできる Recurrent Neural Network（RNN）を用いて，文章の分散表とツイートの前後関係から攻撃性を判定する手法を検討する．

6. 結言

本稿では，Twitter より収集したツイートを用いて学習を行った fastText のモデルを用いて攻撃的な文章のベクトルを獲得し，そのベクトルとのコサイン類似度を用いて文章の攻撃性を評価する手法を提案した．その結果，Twitter から収集したランダムな 50 件のツイートの攻撃性に関するアンケートの結果と，同様のツイートに対する提案手法の攻撃性評価の結果に相関関係があることが確かめられた．具体的には，この二つの点数について，スピアマンの相関係数は 0.7423661，p 値は 6.808e−10 となり，p 値について，6.808e−10<0.01 となったため，有意水準 1%において，アンケートの点数と提案手法の点数には相関関係があるといえる．また，単語単位の一致による検出との比較においては，単語単位の検出では判定することが困難な文章について，正しく判定がなされた例が存在した．しかしながら，逆に単語単位の一致による検出で判定が容易である文章について，正しく判定がなされなかった例も存在した．そのため，短文の攻撃性評価において，提案手法が単語単位の一致による検出より，必ずしも有効であるとは言えない．

謝辞

本研究に際して，様々なご指導を頂きました西村俊二講師に深謝いたします．また，この研究の機会をくださった情報工学科の先生方，そして多くの知識やご指摘を下さいました同研究室の先輩・同期の皆様，アンケートにご回答下さいました皆様に厚く御礼申し上げます．

参考文献

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov:“Enriching Word Vectors with

Subword Information” Transactions of the Association for Computational Linguistics, vol. 5, pp. 135-146, 2017

[2] Tomas Mikolov,, Kai Chen, Greg Corrado, Jeffrey Dean, : “Efficient estimation of word representations in vector space” In Proceedings of Workshop at ICLR 2013

[3] 塚野駿，柴田千尋，政倉祐子，田胡和哉，“ニューラルネット言語モデルによる Twitter 上の発言からの 5 因子モデルに基づく性格分析” 第 78 回全国大会講演論文集，2016，1，pp. 3－4，2016-03-10

[4] 松林圭，五味京祐，古川和祈，尾祐佳，松原良和，日諸マルセロ優次，中村拓哉，山下晃弘，松林勝志，“Twitter 上に投稿された文章に基づく感情推定法とその応用に関する検討”第 78 回全国大会講演論文集 2016，1，pp. 79－80，2016-03-10

[5] 大西真輝，澤井裕一郎，駒井雅之，酒井一樹，進藤裕之，”ツイート炎上抑制のための包括的システムの構築” 人工知能学会全国大会論文集，29，pp. 1-4

[6] 田口雄哉，田森秀明，人見雄太，西鳥羽二郎，菊田洸，“同義語を考慮した日本語の単語分散表現の学習” 研究報告自然言語処理（NL），2017-NL-233 17，pp. 1－5，2017-10-17

[7] Taku Kudo, Kaoru Yamamoto, Yuji Matsumoto: “Applying Conditional Random Fields to Japanese Morphological Analysis” Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004), pp.230-237 (2004)

[8] https://help.twitter.com/ja/rules-and-policies/twitter-api

[9] https://github.com/studio-ousia/mojimoji

[10] https://github.com/neologd/mecab-ipadic-neologd

[11] Mikhail Belkin, Partha Niyogi, :” laplacian eigenmaps and spectral techniques for Embedding and Clustering”Advances in Neural Information Processing Systems 14 (NIPS 2001)

[12] J. MacQueen:“Some methods for classification and analysis of multivariate observations” Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., Vol. 1 (Univ. of Calif. Press, 1967), 281-297

[13] C.Spearman , “The Proof and Measurement of Association between Two Things” The American Journal of Psychology，Vol. 15， No. 1 (Jan., 1904), pp.. 72-101

[14] Ledell Wu， Adam Fisch， Sumit Chopra， Keith Adams , Antoine Bordes and Jason Weston, :“StarSpace: Embed All The Things!” The Thirty-Second AAAI Conference on Artificial Intelligence, AAAI18 - NLP and Machine Learning, 2018

[15] Dheeraj Mekala, Vivek Gupta, Bhargavi Paranjape, Harish Karnick , :“SCDV : Sparse Composite Document Vectors using soft clustering over distributional representations” Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 659–669, Copenhagen, Denmark, September 7–11, 2017

[16] Hitoshi Isahara, Francis Bond, Kiyotaka Uchimoto, Masao Utiyama and Kyoko Kanzaki:”Development of Japanese WordNet”, In LREC-2008, Marrakech., 2008

[17] Taku Kudoh, Yuji Matsumoto, :apanese Dependency Analysis Based on Support Vector Machines , EMNLP/VLC 2000，2000