鉄道トラブルに関する tweet の自動抽出手法

鳥海 心† 宮﨑 太郎‡ 後藤 淳‡ 山田 一郎‡ 八木 伸行†
†東京都市大学 ‡NHK 放送技術研究所
† {g1683108, yagi}@tcu.ac.jp ‡{miyazaki.t-jw, goto.j-fw, yamada.i-hy}@nhk.or.jp

1. はじめに

Twitter は、速報性が高く、ユーザの身の回りで発生している事や話題となっている事について頻繁に投稿される。電車利用者が発信する tweet には、発生したトラブルに関する現場の詳細な情報が含まれており、放送局にとって有益な情報源になる。現在、NHK では tweet から有益な情報をリアルタイムに抽出するためのプロジェクトを立ち上げ、速やかな情報収集への取り組みを進めている[1]。しかし、人手に頼る部分が多く、労力がかかってしまっている。

このような問題を解決するために、Twitter から鉄道トラブルを対象とした情報を抽出する研究を進めている。鉄道トラブルに関する tweet には「読売ランド前で線路冠水！？」のように、路線名を含まないなど、全ての情報が 1 つの tweet で網羅されているわけではない。同じ事象について述べている他の tweet も含めて考慮することによって、より詳細な情報を獲得することが可能となる。

本稿では、複数の tweet を考慮した、鉄道トラブルに関する tweet 収集手法を提案する。提案手法では、まず、路線名を含む tweet から発生場所やトラブル原因など、そのトラブルを特徴的に表しているクエリを拡張し、トラブルに関係する tweet を幅広く収集する。次に、鉄道トラブルについて言及している tweet を学習させた分類器を用いて、抽出した tweet をランキングする。ランキング上位の tweet を順に出力することで鉄道トラブルを表している有用な tweet を獲得する。tweet をランキングする際、SVM(Support Vector Machine)と NN(Neural Network)の 2 つの分類器を準備し、素性に単語の分散表現を用いる。

評価実験では、用意した二つの手法とベースライン手法により出力された tweet を確認し、上位にランキングされた tweet に、ニュースで必要となる情報が含まれているか比較した。その結果、NN による手法の有効性が見られた。

2. 関連研究

近年、Twitter からの情報抽出は多く研究されている。山本らの手法[2]は、電車の遅延情報やスーパーの安売りに関する情報など実生活に役立つ情報を tweet から抽出する手法を提案した。この手法は、実生活情報が掲載されている記事から特徴的な単語を抽出し、生成した専用辞書を用い、実生活を表している tweet を獲得した。しかし、tweet が実生活情報であるか否かの判定の際、辞書に登録された単語のみを使用しているため、tweet の単語数に依存し、誤った tweet を実生活情報であるという判定をしてしまうことがある。

また、鉄道トラブルを対象に tweet から情報抽出した土屋らの手法[3]は、鉄道トラブルに関する 3 つのタスクに取り組んでいる。一つは、鉄道トラブルの発生時間及び復旧時間を検出し、同時に、全線見合わせ、一部見合わせなどのトラブルの状態を検出した。二つ目は、トラブルが一定時間以上長引くか否か予測した。三つ目はトラブルが他の路線に影響を及ぼすか予測した。この手法では、路線名を含む tweet のみを対象としているため、トラブルに関して言及しているにもかかわらず、路線名を含んでいない tweet を考慮していない。

3. 提案手法

提案手法は、図 1 のように①対象トラブルに合わせたクエリ拡張、②分類器を用いた tweet のランク付け、の 2 段階で処理し、トラブルを表している有用な tweet を獲得する。以下で各処理の詳細を説明する。

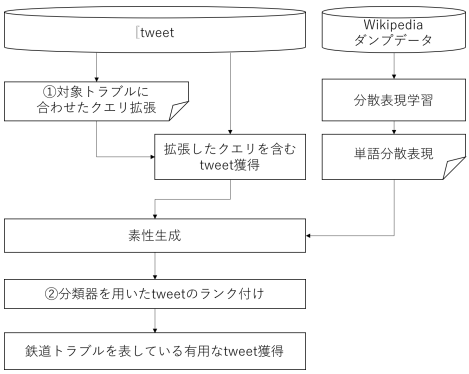

図 1 提案手法の概要

3.1 対象トラブルに合わせたクエリ拡張

対象トラブルごとに設定した期間内の全 tweet から、路線名をクエリとしたキーワードマッチングで tweet を抽出する。抽出した全ての tweet を形態素解析し、tweet 内の助詞、助動詞を除く全単語分の TFIDF を計算する。その後、抽出した全 tweet に出現する単語の TFIDF を足し合わせることにより、トラブルを表す特徴的な単語を順にランキングする。ランキングした単語の上位 15 位に含まれる名詞を抽出し、人手によりストップワードを削除したものを拡張したクエリとする。2016 年 9 月 2 日の「田園都市線」を含む tweet を対象にクエリ拡張した例を表 1 に示す。この日に東急田園都市線の用賀駅で窓ガラス破損のトラブルが発生している。

表 1 拡張クエリ例

• 田園都市	• 運転見合わせ
• 用賀駅	• 用賀
• 窓ガラス	• お客様
• トラブル	• 同士

3.2 分類器を用いた tweet のランク付け

3.1 節で拡張したクエリを含む tweet を分類器を用いてランキングし、鉄道トラブルを表している有用な tweet を獲得する。

まず、拡張したクエリのいずれかを含む設定期間内の tweet を抽出する。次に、分類器により、それぞれの tweet にスコアを付け、スコアに基づいて抽出した tweet をランキングする。分類器の学習には、放送局で収集している鉄道トラブルに関する tweet を正例として利用する。表 2 に、学習に使用した tweet の正例を示す。

分類器には SVM と NN の二つを用意し、SVM は分離平面からの距離、NN では出力の重みで tweet をランキングする。

分類器に入力する素性には、単語の分散表現を用いた。単語の分散表現は単語の表層を用いず、単語の出現する文脈により多次元の特徴を表現できることから、近年注目を浴びている。単語の分散表現による素性は、判定の対象となる tweet に出現するすべての単語の分散表現の和をとり、正規化したものを用いる。

$$v^{(i)} = \frac{\sum_{w \in W} v(w)^{(i)}}{|W|}$$

ここで、$v^{(i)}$は i 番目の素性を、W は tweet に出現する単語の集合を、$v(w)^{(i)}$は単語 w の分散表現の i 番目の要素をそれぞれ表す。

Skip-gram により作成されたベクトルの和は、意味的な足しあわせを表現できることが報告されている[4]。tweet に出現するすべての単語の分散表現の和は、tweet 全体の意味を考慮した特徴になっていると考えられる。

tweet をランキング上位から順に出力することで、多数の tweet を見ることなく、ニュースに必要な全ての情報を把握することができる。表 3 に本手法を用いて上位にランキングされた tweet の例を示す。

表 2 学習に使用した tweet の正例

- 田園都市線内で停電だって
- 中央線 架線にビニールで運転見合わせ
- 京急が止まってる
- 電車停電なう
- 東小金井駅で人身事故が起きた

表 3 上位にランキングされた tweet

- 今朝の田園都市線は用賀駅で電車の窓ガラスが割れたそうで、止まっている!
- 電車の窓ガラス破損とか何事
- 用賀駅で客同士のトラブルって…
- ついに田園都市線、準急で窓ガラス破損、マジか…

4. 評価実験

提案手法の効果を確認するため、評価実験を行った。提案手法により出力された tweet

を 1 名の被験者により評価をした。評価方法は、いかに少ない tweet でニュースに必要な情報を獲得できるかを評価するため、出力した tweet を 1 位から順に確認し、路線名、発生場所、鉄道状況、トラブル原因の 4 つの情報をそれぞれ何番目の tweet で獲得することができるか確認した。対象とする tweet の期間は、トラブル発生時刻からオフィシャル tweet を含む鉄道会社の公式発表までとする。公式発表時間が確認できなかったトラブルは、トラブル発生後 10 分以内を期間とする。以下で実験条件について述べる。

4.1 実験条件

今回、2016 年に発生した 5 つの鉄道トラブルを実験の対象とした。表 4 に鉄道トラブルの概要を示す。

クエリ拡張に用いた形態素解析には MeCab[5]を使用した。単語をランキングするための IDF の計算には、Wikipedia の 2016 年 9 月のダンプデータを用いた。

3.2 節で tweet をランキングするために使用した SVM には SVM-Light[6]を使用し、多項式カーネルにより判定した。NN の実装は Chainer[7]を使用し、入力層、中間層、出力層の 3 層でネットワークを構築した。入力層は 200 次元、中間層は 100 次元、出力層は 2 次元に設定した。単語の分散表現は Word2Vec[8]を用い、IDF の計算と同様、Wikipedia のダンプデータから計算した。

表 4 実験に用いた鉄道トラブル概要

日時	路線	発生場所	状況	原因
7/14	小田急線	読売ランド前駅	運転見合わせ	線路冠水
8/16	田園都市線	桜新町駅	運転見合わせ	信号機故障
8/17	中央線	三鷹駅	運転見合わせ	人身事故
9/2	田園都市線	用賀駅	運転見合わせ	ガラス破損
9/26	日比谷線	霞ヶ関駅	運転見合わせ	発煙

4.2 ベースライン手法

提案手法との比較に用いるベースライン手法として、単語の重み付けに一般的に用いられる BoW(Bag of Words)を用いた。tweet に出現する 3.1 節の手法により拡張したクエリのみを足し合わせることで tweet のスコアを求め、上位から順位出力する。

4.3 結果

評価実験の結果を表 5 に示す。表内の数値は各情報を獲得できた tweet の順位を表し、全情報は全ての情報を獲得できた tweet の順位を示す。“-”はスコア上位 50 位以内にその情報が含まれない場合である。それぞれ対象とした期間内で路線名を含む tweet 数は「7/14 小田急線」は 122 個、「8/16 田園都市線」は 8 個、「8/17 中央線」は 24 個、「9/2 田園都市線」は 38 個、「9/26 日比谷線」は 10 個である。結果より、提案手法を用いることで、ランキング上位の tweet にニュースに必要となる情報が含まれていることがわかった。そのうち、NN による手法では、3 つの事例においてより少ない tweet 数で全てのトラブル情報を獲得できた。

表 5 実験結果

	手法	路線	場所	状況	原因	全情報
7/14 小田急線	BoW	3	-	3	1	-
	SVM	**1**	**2**	**1**	**2**	**2**
	NN	1	40	1	3	40
8/16 田園都市線	BoW	1	-	4	-	-
	SVM	3	-	6	-	-
	NN	3	-	21	-	-
8/17 中央線	**BoW**	**1**	**1**	**2**	**1**	**2**
	SVM	2	1	5	1	5
	NN	**1**	**1**	**2**	**1**	**2**
9/2 田園都市線	BoW	4	2	1	4	4
	SVM	4	4	2	1	4
	NN	**2**	**2**	**1**	**2**	**2**
9/26 日比谷線	BoW	1	3	3	3	3
	SVM	1	2	2	2	2
	NN	**1**	**1**	**1**	**1**	**1**

3.4 考察

評価実験により、NN を用いた手法が有効であることがわかった。しかし、トラブル原因が台風やゲリラ豪雨など他の路線にも影響

を及ぼす場合、クエリを拡張することで他の路線について言及している tweet も多数拾ってしまう。その際、SVM を含む分類器による手法では、対象としたトラブルに関する tweet を有効にランキングすることができない。一方、ベースラインに用いた BoW による手法では、対象トラブルに関する tweet を獲得することが可能である。このため、BoW と分類器を用いる手法を組み合わせることで、より対象トラブルに関する有用な tweet を上位にランキングできると考えられる。

今回の評価実験で情報獲得できなかった「8/16 田園都市線」のトラブルは、信号機故障が原因であり、公式発表があるまでは現地にいた人にも原因がわからなかったと考えられる。トラブル発生直後に投稿された tweet には、トラブルの詳細情報を含む tweet が存在せず、「田園都市線、止まった」など簡単な鉄道状態やネガティブな tweet が多数見られた。そのため、有効にクエリ拡張をすることができず、トラブルに関する tweet をうまく集めることができなかった。このような事例のトラブルに関しては、新たな手法を考える必要がある。

また、評価実験の際に、いかに少ない tweet で詳細な情報を獲得できるかを評価するために、獲得した上位 10 位までの tweet を被験者に提示し、手作業により要約を作成した。「9/2 田園都市線」のトラブルについての結果を表 6 に示す。NN を用いた手法では、「準急」や「渋谷方面上り線の一部で運転見合わせ」、「半蔵門線にも遅れの見込み」などより詳細な情報まで獲得できた。これより、NN による手法が最も多くの情報を抽出することができ、有効性を確認した。

表 6 要約作成の結果

手法	要約
BoW, SVM	田園都市線用賀駅で乗客同士のトラブルにより電車の窓ガラスが破損、運転見合わせ。
NN	田園都市線用賀駅準急で乗客同士のトラブルにより電車の窓ガラスが破損。渋谷方面上り線の一部で運転見合わせ。半蔵門線にも遅れの見込み。

5. おわりに

本稿では、鉄道トラブルに関する一報の収集のために、tweet から素早く詳細な情報を獲得する手法について提案した。

まず、鉄道トラブルに関する詳細な情報獲得のために、路線名から発生場所、トラブル原因などトラブルに関係するクエリを動的に拡張した。次に、鉄道トラブル tweet を幅広く集め、トラブルを表している有用な tweet を分類器によりランキングし、抽出した。分類器には SVM と NN の二つを用いて比較した。実験では、提案手法により上位にランキングされた tweet にニュースに必要な情報が含まれるかを評価した。その結果、NN を用いる手法が有効であることを確認した。

今後、より有用な tweet を獲得するために、BoW と分類器を組み合わせる手法や、語順を考慮することができる Recurrent Neural Network を用いた手法を検討する。

本研究の一部は、JSPS 科研費 25280036 の助成を受けたものです。

参考文献

[1] 足立 義則, “震災ビッグデータからソーシャルリスニングへ,” 放送メディア研究, No.11, pp.290-293, (2014).

[2] 山本 修平ほか, “Twitter からの実生活情報の抽出法の提案,” DEIM Forum 2012, B3-2 (2012).

[3] 土屋 圭ほか, “マイクロブログを用いた鉄道の運行トラブル発生期間および付帯情報の抽出,” DEIM Forum 2012, F3-4 (2012).

[4] Tomas Mikolov, et al., “Distributed representations of words and phrases and their compositionality,” Advances in neural information processing systems, (2013).

[5] Taku Kudo, et al., “Applying Conditional Random Fields to Japanese Morphological Analysis,” in Proceedings of EMNLP 2004, pp. 230–237, (2004).

[6] Thorsten Joachims, “Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning,” MIT-Press, (1999).

[7] Seiya Tokui et al., “Chainer: a Next-Generation Open Source Framework for Deep Learning,” in Proceedings of NIPS 2015 workshop, (2015).

[8] Tomas Mikolov, et al., “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, (2013).