

Twitter からの鉄道トラブル自動検知システム

鳥海 心† 宮崎 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

近年, 放送局では Twitter などのソーシャルメディアから有用な情報を収集する取り組みを進めている[1]. ソーシャルメディアには, ユーザの目の前で起こった事件や事故が投稿されるため, ニュース性のある情報の第一報を得ることができる. しかしながら, 大量の Tweet から必要な情報を取得するためには, 収集や分析に大きな労力が必要である. そこで, 我々は, 現場の労力を削減するために鉄道トラブルに関する Tweet の自動抽出手法について研究を進めている[2][3].

ニュース制作現場での実運用を考えると, 時々刻々と投稿される Tweet から自動的にトラブルを検知する処理は不可欠である. そこで本稿では, Twitter の投稿から何らかのイベントが起こったことを検出する機構を備えた, トラブル検出及びトラブル情報を含む Tweet の獲得を自動的に行うシステムを提案する. 鉄道トラブル検出手法の評価と, トラブル検出から鉄道トラブルに関する Tweet をランク付けして有益な Tweet を自動獲得する提案手法の総合評価を行った結果, 鉄道トラブル検出手法の有効性を確認し, 提案手法全体としての評価でも自動で鉄道トラブルを検知し, 鉄道トラブルに関連する有益な Tweet を抽出できることを確認した.

2. 関連研究

Twitter から鉄道運行状況に関する情報抽出は多く研究されている. 新井ら[4]は鉄道遅延を対象に, 鉄道運行情報を推定する手法を提案している. この手法は, 複数のルールを適用し, 運行情報に関する Tweet を抽出することで速報性と情報の正確性を向上させている. 土屋ら[5]は鉄道トラブルの発生時間及び復旧時間を検出し, 同時にトラブル状態を検出している. 加えてトラブルが一定時間以上長引くか否かの予測と, トラブルが他路線に

影響を及ぼすかの予測を行っている. これらの手法は路線名を含む Tweet を対象に解析しているため, 路線名を明記していない Tweet から情報を獲得することは難しい. 本稿で提案する手法では対象トラブルに合わせたクエリ拡張を自動的に行うことで, 路線名などが示されていない Tweet を考慮することができ, より幅広い情報獲得が可能である.

3. 提案手法

図 1 に提案手法の概要を示す. 本手法は, ①鉄道トラブル検出手法, ②鉄道トラブルに関する有益な Tweet 抽出手法の 2 つの処理を自動で行い, トラブル検出及び対象の鉄道トラブルを表している有用な Tweet を獲得する.

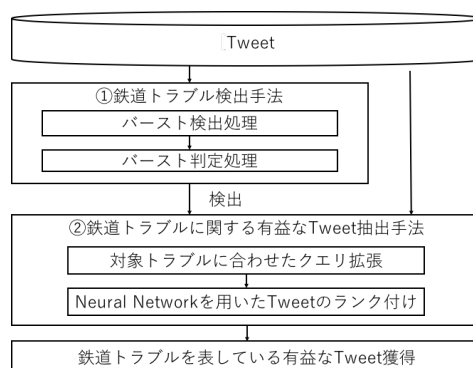


図 1 提案手法の概要

3.1 鉄道トラブル検出手法

図 2 に鉄道トラブル検出手法の詳細を示す. まず, 特定の路線名を含む Tweet の投稿に対し, 蝦名ら手法[6]を用い, バースト検出を行う. その後, Neural Network(NN)を用いたバースト判定手法により検出したバーストが鉄道トラブルに関係するか否かを判定する.

蝦名らのバースト検出手法では, Aggregation pyramid と呼ばれるデータ表現を用い, リアルタイムにバーストを検出する. Tweet の投稿される時間間隔からバースト検出するため, 投稿がない期間の計算を削減できる. 提案手法では鉄道路線名を含む Tweet

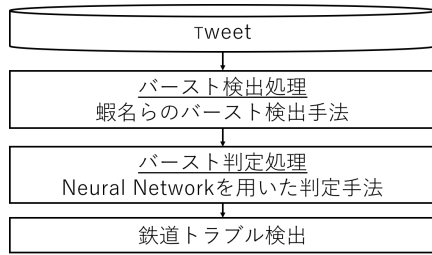


図 2 鉄道トラブル検出手法の概要

を対象としたバースト検出を行うが、通常時には対象となる投稿がない場合が多いため、その時間の計算量を削減できる本手法が今回の解析には適していると考えた。提案手法では、バースト解析時間の最小単位を 1 分とし、バースト判定された時間をトラブル発生時間とする。

検出したバーストが、鉄道トラブルに関連するか判定するために、バースト検出に用いられた Tweet を予め鉄道トラブルに関係するか否か判定するように学習させた NN に入力する。過半数の Tweet に対して鉄道トラブルに関係すると判定した場合、対象バーストを鉄道トラブルによるバーストとみなす。NN 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、ランダムサンプリングした Tweet を負例とする。素性には Tweet に含まれる単語の分散表現の加算平均を用いる。

3.2 鉄道トラブルに関する有益な Tweet 抽出手法

先行研究[2]で提案した手法を用いる。まず、バースト検出の原因となった Tweet からクエリ拡張する。拡張したクエリで Tweet を検索し、獲得した Tweet を鉄道トラブル検出に適用した NN を用いてトラブルに関連するか否かの判定とランク付けを行う。

対象トラブルに合わせたクエリ拡張では、バースト検出に用いた Tweet に含まれる助詞、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、トラブルを表す特徴的な単語を順にランキングする。ランキング上位 15 位に含まれる名詞を抽出し、ストップワードを削除したものを拡張クエリとする。

トラブルに関連するか否かの判定とランク付けでは、拡張したクエリのいずれかを含む

Tweet をキーワードマッチングにより抽出する。抽出する Tweet は、バースト判定に用いた Tweet の投稿時間内のものとする。次に、抽出した Tweet を鉄道トラブル検出処理に用いた NN に入力する。それぞれの Tweet における NN の出力の重みをスコアとして求め、スコアに基づいて Tweet をランキングする。

4. 評価実験

提案手法の効果を確認するため、「鉄道トラブル検出手法の評価」と「提案手法の総合評価」の実験を行った。

4.1 実験条件

2017 年 7~10 月に小田急線公式 Tweet アカウントが「不通区間」を含む Tweet を投稿した 18 件のトラブル(DATA2017)と我々の先行研究[2]で評価データとして扱った 2016 年 7/14, 8/16, 9/02, 9/26 に起きた 4 件のトラブル(DATA2016)を実験データとして用意した。バーストアルゴリズムのパラメータは DATA2017 の結果が最適となるよう調整した。バースト判定やクエリ拡張に用いた形態素解析には MeCab[7]を使用した。単語ランキングに用いた IDF の計算には、Wikipedia の 2016 年 9 月のダンプデータを用いた。NN の実装は Chainer[8]を使用し、入力層、中間層、出力層の 3 層からなる Feedforward Neural Network 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[9]を使い、Wikipedia のダンプデータから計算した。

4.2 鉄道トラブル検出手法の評価実験

鉄道トラブル検出手法の性能を確認するために、「バースト検出処理の性能評価」と「バースト判定処理の性能評価」を行った。

本実験では、DATA2017 を対象とした。入力データには、トラブルに関する第一報の公式 Tweet 投稿時間の前後 30 分とその直前の 3 時間、直後の 1 時間に投稿された Tweet を用いた。例えば、12:30 に公式 Tweet が投稿された場合、9:00~14:00 の Tweet を対象とした。

まず、鉄道トラブルの発生を検知できるか確認するため、対象トラブルによる第一報のバーストを検出可能か評価した。また、時間により変化する運行状況の情報を抽出するためにはトラブル発生から復旧するまでに検出

したバースト全てを鉄道トラブルに関連するか評価する必要がある。そこで、バースト検出手法により正しくバーストが検出できたかを確認する実験を行った。Tweetを1分ごとに時間幅で区切り、各時間幅ごとにトラブル情報を含むTweetが過半数含まれる場合を鉄道トラブルによるバーストであったと定義し、適合率と再現率で評価した。被験者一人がバーストと判定されたTweetについて、鉄道トラブルに関するバーストであるかを評価し、鉄道トラブルに関連するTweetが過半数を超えた場合、そのバーストを鉄道トラブルによるものとした。

DATA2017の全18件のトラブル中、14件においてバーストの検出を確認した。そのうち、13件は鉄道トラブルの第一報を表すバーストを検出できた。表1に、バースト検出処理により検出したバーストのトラブルごとの適合率を示す。14件を合計した場合の適合率は0.826であり、提案手法に用いたバースト検出処理の有効性を確認した。

表1 バースト検出処理の評価結果

日付	バースト数	適合率
7/1	14	0.785
7/5	18	1.000
7/17	16	1.000
7/18	0	-
7/30	4	1.000
8/1	40	0.975
8/4	3	1.000
8/11	6	1.000
8/21	7	1.000
8/27	0	-
8/30	0	-
9/1	15	0.933
9/3	5	1.000
9/5	31	0.935
9/10	14	1.000
10/1	0	-
10/23	58	0.362
10/30	22	1.000
合計	253	0.826

次に、バースト判定処理を組み合わせることで、無関係なバーストを検出せずに鉄道トラブルに関するものだけを検出できるか評価した。バースト検出手法により得られた全バーストに対して、バースト判定処理を行った。評価方法は適合率と再現率を求め、バースト検出手法のみとの結果と比較した。

表2にバースト判定処理を用いた際の結果を示す。全てのバーストを合計した結果では、適合率が0.828、再現率0.942であり、適合率の大きな向上は確認できなかった。

表2 バースト判定処理の評価結果

日付	バースト数	適合率	再現率
7/1	11	1.000	1.000
7/5	18	1.000	1.000
7/17	16	1.000	1.000
7/30	4	1.000	1.000
8/1	31	0.967	0.769
8/4	3	1.000	1.000
8/11	6	1.000	1.000
8/21	7	1.000	1.000
9/1	15	0.933	1.000
9/3	5	1.000	1.000
9/5	30	0.933	0.965
9/10	13	1.000	0.928
10/23	58	0.362	1.000
10/30	22	1.000	1.000
合計	239	0.828	0.947

4.3 提案手法の総合評価実験

鉄道トラブル検出から有益なTweet抽出まで自動で行った場合の総合的な提案手法の評価を行った。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、運行状況、原因の4つの情報をそれぞれ何番目のTweetで獲得可能か確認した。

実験には、先行研究[2]で評価したDATA2016を対象とし、比較対象に先行研究の実験結果を用いる。先行研究では、トラブル検出のタイミングとしてトラブル発生時刻を与え、そこから鉄道会社の公式Tweet投稿時間内のTweetを用いてクエリ拡張している。提案手法では、バースト検出アルゴリズムにより検出した最初のバーストを用いた。

表3に結果を示す。表内の数値は各情報を獲得できたTweetの順位を表す。“-”はスコア

表3 提案手法の実験結果

日	手法	路線	場所	状況	原因
7/14	提案手法	1	10	1	3
	先行研究	1	40	1	3
8/16	提案手法	1	-	15	-
	先行研究	3	-	21	-
9/2	提案手法	1	2	4	2
	先行研究	2	2	1	2
9/26	提案手法	1	1	1	1
	先行研究	1	1	1	1

上位 50 位以内にその情報が含まれない場合である。鉄道トラブル検出を自動で行った提案手法と人手で与えた先行研究と同等の結果であることがわかった。よって、提案手法を用いることで、鉄道トラブル検出からトラブルに関連する Tweet の獲得を自動で行うことが可能であると言える。

5. 考察

鉄道トラブル検出手法の評価実験において、バースト判定処理を加えた結果、鉄道トラブルを表すバーストを無関係のものと判定することがあった。誤判定をしたバーストを確認した所、鉄道トラブルを表す Tweet を含んでいるが、過半数を超えていないために無関係と判定している。このような誤判定を防ぐため、バーストを判定する Tweet 数の閾値を検討する必要がある。

10/23 のトラブルは適合率 0.36 であり、他トラブルと比べ、低い結果であった。トラブル原因は人身事故であるが、台風のため他路線が運転を見合わせていた。そのため、「小田急線」を含み、他路線の運行情報に言及した Tweet が多く投稿され、それらが原因となり、バーストを誤検出してしまった。このような、他路線にも影響を及ぼすイベントに対しては新たな手法を考慮する必要がある。また、10/23 のトラブルで検出されたバーストは全体の 4 割(58/239)を占めるため、結果に与える影響が大きい。そこで、このトラブルを除き、再評価した結果、バースト判定を加えることで、適合率 0.98、再現率 0.94、バースト検出処理のみでは適合率が 0.96 のため、バースト判定により正確なバーストを検出できることを確認した。

先行研究[2]の 7/14 の結果では場所の検出が 40 位であるのに対して、今回の提案手法の総合評価では 10 位で獲得できている。提案手法ではバースト検出により対象の時間をより絞った Tweet に対象が限定できるため、抽出した Tweet 数が先行研究より少なく、情報が埋もれなかったためと考える。

鉄道トラブルの検出時間と公式 Tweet 投稿時間を比較した結果、DATA2016 の 4 件すべてにおいて公式 Tweet よりも早くトラブルを検出することができた。また、DATA2017 でバースト検出できた 14 件のトラブルでは、8

件が公式 Tweet よりも早く検出できた。

6. おわりに

本稿では、鉄道トラブル検出から関連する有益な Tweet 抽出まで自動で行う手法を提案した。鉄道トラブル検出手法は、蝦名らの手法を用い、路線名を含む Tweet のバーストを検出する。その後、検出したバーストが鉄道トラブルに関連するか否か判定し、鉄道トラブルによるバーストを検出する。実験では、鉄道トラブル検出手法の性能評価と、提案手法の総合評価を行った。その結果、鉄道トラブル検出手法では、正確にトラブルに関連するバーストを検出できることがわかった。総合評価では、自動でトラブル検出から有益な Tweet 抽出まで行うことができることを確認した。今後、本手法がトラブルの時系列変化を獲得する手法を検討していく。

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] 鳥海心ほか, “鉄道トラブルに関する tweet の自動抽出手法,” 第 23 回言語処理学会年次大会発表論文集, D4-1, pp.418-421, (2017).
- [3] 鳥海心ほか, “時系列推移を考慮した鉄道トラブルに関わる tweet 抽出手法の検討,” FIT2017(第 16 回情報科学技術フォーラム), E-013, (2017).
- [4] 新井誠也ほか, “速報性と正確性の向上を図った Twitter からの鉄道運行情報検出システム,” DEIM Forum 2015, B5-5, (2015).
- [5] 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の発生検出および付帯情報の検出.,” DEIM Forum2014, B3-2, (2014).
- [6] 蝦名亮平ほか, “リアルタイムバースト検出手法の提案,” 日本データベース学会論文誌, Vol.9, No.2, (2010)
- [7] Taku Kudo, et al., “Applying Conditional Random Fields to Japanese Morphological Analysis,” in Proceedings of EMNLP 2004, pp. 230–237, (2004).
- [8] Seiya Tokui et al., “Chainer: a Next-Generation Open Source Framework for Deep Learning,” in Proceedings of NIPS 2015 workshop, (2015).
- [9] Tomas Mikolov, et al., “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, (2013).