

Twitterからの有用情報抽出のための 学習データのマルチクラス化

宮崎 太郎^{1,a)} 鳥海 心² 武井 友香¹ 山田 一郎¹ 後藤 淳¹

概要：Twitterなどのソーシャルメディアの情報は、現在ではニュース制作者にとって重要な取材源となっている。NHKではソーシャルメディアを活用し、ニュースに役立つ情報を迅速に抽出するためのプロジェクトを立ち上げ、積極的に活用しているが、人手に依るところが多く、その労力は大きい。そこで我々は、ニュース制作に役立つ情報をソーシャルメディアから自動で抽出する研究を進めている。再帰型ニューラルネットワークを用い、ニュースに役立つ情報を自動で抽出する手法の研究を進めてきたが、従来は「ニュースに役立つか役立つかないか」という2値分類をタスクとしてきた。しかし、実際の「ニュースに役立つ tweet」には「火事」「事故」など多様な情報が含まれており、それらを一つに括って扱うのは性能の劣化につながることを考えられる。そこで、本稿では「ニュースに役立つ tweet」を分野ごとに細分化して学習する手法を提案する。学習データについて、人手でニュースに役立つ tweet をニュースの分野ごとに分類した場合には情報抽出の精度がF値の最大で0.744となり、ニュースに役立つ tweet を一括りにした場合と比較して0.033ポイントの性能向上が得られた。また、自動で分類した場合にはF値が最大で0.732となり、同じく0.021ポイントの性能向上が得られた。本稿の貢献として、学習データを人手、または自動で細分化して学習することで、識別の性能が向上することを明らかにしたことが挙げられる。

Classifying Training Data for Extracting Important Tweets

MIYAZAKI TARO^{1,a)} TORIUMI SHIN² TAKEI YUKA¹ YAMADA ICHIRO¹ GOTO JUN¹

1. はじめに

近年、twitterなどのソーシャルメディアには、数多くの情報が投稿されている。この中には、事件や事故の現場に偶然居合わせた人からの、写真や動画付きの目撃情報も多く含まれている。これらを活用することで、素早い情報収集が可能となることから、現在ではニュース制作者にとっての重要な取材源のひとつになっている。NHKでもソーシャルメディアを活用し、ニュース制作に役立つ情報を迅速に抽出するためのプロジェクトを立ち上げており、既に実際の報道番組などでその成果が活用されている [1]。ここでは、タイムライン上の情報を、複数の検索クエリを組

み合わせてフィルタリングした上で、人手で情報を選別している。しかしこの作業には大きな労力がかかり、現場への負担が大きい。また、検索クエリでのフィルタリングでは、例えば「明日の試験の勉強していないし、電車止まらないかなあ」のようなニュース制作には役立つ情報も、「電車」と「止まる」のキーワードを含むために抽出されてしまう。そこで我々は、検索クエリに頼らずにより精度良くニュース制作に役立つ情報を自動で抽出するための技術の研究に取り組んでいる [2], [3], [4]。

これまで、我々はその tweet がニュースに役立つ／役立つかないの2値分類をタスクとしていた。しかし、ニュースに役立つ tweet には、「火事」や「自動車事故」、「異常気象」など様々な分野のものが含まれている。実際のニュース制作現場での利用を考えると、ニュースの分野を表すラベルも合わせて出力したほうがより使いやすい。また、分野が異なれば tweet の内容も大きく異なるが、これらを一つの

¹ NHK 放送技術研究所
NHK Science and Technology Research Laboratories

² 東京都市大学
Tokyo City University

^{a)} miyazaki.t-jw@nhk.or.jp

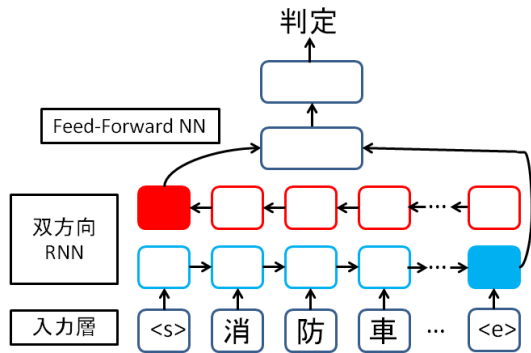


図 1 RNN による tweet の判定

「ニュースに役立つ」という括りでまとめていたため、それが性能の劣化に繋がっていたことが考えられる。そこで本稿では、tweet がニュース制作に役立つか判定する際に、ニュースの分野を表すラベルも合わせて推定することを考える。これにより、アプリケーションとして、抽出した情報がどの分野のニュース制作に役立つかのラベルをユーザに提供することができるだけでなく、ニュースに役立つ情報を含む tweet であるかどうかの識別の性能向上も見られた。本稿の貢献としては、学習データに用いるラベルを手動または自動で細分化することにより、手軽に性能が向上できることを明らかにしたことが挙げられる。

2. Tweet 判定手法

Tweet がニュースに役立つかどうかの判定と分類には RNN を用いる。本稿では、[4] で提案した手法をベースとしているが、従来では「その tweet がニュース制作に役立つかどうか」の 2 値分類をタスクとしていたが、本稿では正例、負例をそれぞれマルチクラスに分類する点が異なる。

2.1 RNN を用いた tweet の判定

はじめに、tweet を RNN に入力し、tweet 全体の意味を表すベクトル表現を得る。一般に、ソーシャルメディアは気軽に投稿されることから、口語調で書かれることが多くかつ、略語やスラング、絵文字などが多く出現する。そのため、一般の形態素解析器では、うまく単語単位に分割できない場合も多い。また、投稿の内容が多岐にわたるため、出現する語彙数が膨大なものになる。そのため、従来、単語単位でテキストデータを扱う手法が一般的であったが、ソーシャルメディアを対象とする場合には文字単位で扱うことで良い性能が得られるという報告がされている [5], [6]。本稿でも tweet を文字ごとに RNN に入力することとする。

図 1 に、RNN を用いた tweet の判定手法の概要を示す。ベクトル表現を得るための RNN は順方向と逆方向の 2 つを用い、該当の tweet をそれぞれの RNN に順方向、逆方向に入力する。全ての文字の入力後、終端記号を入力した時点でのそれぞれの RNN の内部状態を表すベクトルを結合したものを、tweet 全体の意味を表すベクトル表現として用

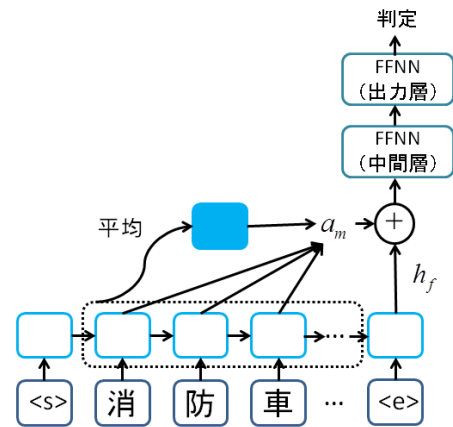


図 2 MeanVector アテンションの計算
 (順方向 RNN での計算の例)

いる。得られたベクトル表現を用い、2 層の Feed-Forward Neural Network によりニュース制作者に有用であるかどうか判定する。

2.2 アテンションメカニズムの導入

アテンションメカニズムは、近年、機械翻訳やキャプション生成にも多く用いられる手法で、入力データのうちの重要な部分への重み付けに用いることができる [7], [8]。本稿では、このアテンションメカニズムを利用し、入力 tweet 中の重要部分に重み付けすることで、判定性能の向上を目指す。アテンションの計算方法には MeanVector アテンションを用いる。MeanVector アテンションは、今回のタスクにおいて良好な性能が得られることがわかっている [4]。

MeanVector アテンションは、tweet を文字ごとに入力した際に得られる、各文字ごとの RNN の状態の平均を用いてアテンションを計算する。以下では、順方向 RNN の場合で説明するが、実際には順方向、逆方向それぞれの RNN でアテンションを計算し、判定に用いている。

MeanVector アテンションの概要を図 2 に示す。Tweet を文字ごとに RNN に入力した際の、 t 番目の文字まで入力した RNN の状態を \bar{h}_t とすると、 t 番目の文字に与えるスコア $score_t$ は以下のように求められる。

$$score_t = h_m^T \bar{h}_t \quad (1)$$

$$h_m = \frac{\sum_{t'} \bar{h}_t}{|t'|} \quad (2)$$

なお、 t' は対象の tweet に出現するすべての文字の集合を表す。このスコアを用い、各文字の重み W_t を求める。

$$W_t = \frac{\exp(score_t)}{\sum_{t'} \exp(score_{t'})} \quad (3)$$

$$a_m = \sum_{t'} W_t h_t \quad (4)$$

得られた MeanVector アテンションのベクトル a_m と、終端記号を含む全文字を入力した後の RNN の状態 h_f との

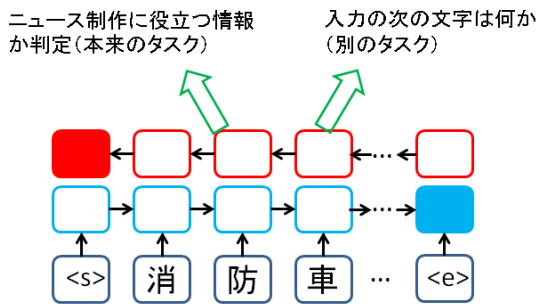


図 3 マルチタスク学習の概要

和を特徴量として用い、tweet がニュース取材に役立つかどうかを判定する。判定には2層の Feed-Forward Neural Network を用いる。

これにより、tweet 全体の文意に近い部分に高い重みを与えることが可能となり、分類の性能が向上することが考えられる。

2.3 マルチタスク学習の導入

ニューラルネットワークを用いた手法では、マルチタスク学習と呼ばれる、一つのモデルを複数のタスクで学習することでより汎用的なモデルを作成する手法により性能が向上することが報告されている [9], [10]。本稿でも、マルチタスク学習を導入することで、性能の向上を試みた。

本稿では、本来のタスクである tweet がニュース取材に役立つかどうかの判定に加えて、入力文字列の次の文字を予測するタスク、すなわち文字単位のニューラル言語モデルを学習することでマルチタスク学習した (図 3)。これにより、新たなデータを準備することなくマルチタスク学習ができる。

入力層と双方向 RNN については2つのタスクで共有し、出力層をタスクごとに使い分けることとした。モデルの学習時には、まず別のタスクを用いて学習する。この結果を初期モデルとして用い、本来のタスクで学習する。

2.4 判定のマルチクラス化

我々は、これまで出力を「ニュース制作に役立つ／役立つかない」の2値分類としていた。しかし、ニュース制作に役立つ tweet には「火事」に関するもの、「自動車事故」に関するものなど様々なものが含まれる。それぞれで内容が大きく異なることから、ニュースの分野ごとに分けてモデルを学習したほうが、より性能が向上することが考えられる。

マルチクラス化のためには、学習データをマルチクラス化する必要がある。今回使用する学習データは、正例には人手でニュース制作に役立つものを用い、負例にはランダム抽出した tweet を用いている (詳細は 3.1 節参照)。学習データ中の正例は数が少なく、内容もある程度の一貫性があるため分類がしやすい。また、ニュースの内容に応じたクラスを出力するためには人手でラベルを与える必要があ

るため、正例については人手で分類することが望ましい。それに対し、負例は数が多く、雑多な内容のものが含まれているため人手での分類が難しい。そこで、本稿では、マルチクラス化のために、正解データ中の正例は人手で分類し、負例は自動で分類することを考える。

負例の分類は tweet 全体の分散表現を獲得したうえで、クラスタリングする。tweet 全体の分散表現の獲得には以下の2種類の手法を用い、性能を比較する。

Word2Vec による分散表現 Word2Vec[11]により事前に学習した単語ごとの分散表現を用いる。Tweet 中に出現する各単語の分散表現の加算平均を tweet 全体の分散表現とする。

RNN による分散表現 まず、本稿で述べた手法で負例を1クラスとした場合について学習する。そこで得られたモデルに tweet を入力した場合の、Bi-RNN の状態を tweet 全体の分散表現とする。

Word2Vec による分散表現は、事前に用意した大規模なデータにより学習が可能であるが、単語単位での分散表現であるため、tweet のように単語分割が難しい場合に性能の劣化が予想される。RNN による分散表現は、学習には時間がかかるが、文字単位でその出現順も含めた分散表現が得られるため、tweet を対象とした今回のタスクでは扱いやすい。また、今回のタスクに合わせた学習の結果を利用しているため、これに合わせた分散表現が獲得できるものと考えられる。

3. 評価実験

3.1 実験条件

学習データとして、正例には報道現場で実際に番組制作に使用した tweet を、負例にはランダム抽出した tweet を用いた。評価データとしては、報道現場での使用を想定し、実際に報道現場で用いている検索クエリによりフィルタリングした tweet からランダムに抽出し、1名の評価者によりそれぞれの tweet について番組制作に役立つまたは役立つかないのラベルを付与したものを用いた。データ量を表1に示す。

マルチクラス分類のために、「番組制作に役立つ tweet」については、ニュースの種別ごとに細分化したラベルも付与した。ニュース種別は23種類に分けている。その内訳と種別ごとのデータ数を表2に示す。今回は正例について同じ23種類に自動で分類したものも用意し実験により性能を比較する。

負例の分類に用いる Word2Vec の学習データには2016年9月の wikipedia のダンプデータを用いた。ダンプデータの単語分割には MeCab*1を、クラスタリングには Repeated Bisection 法を用いたクラスタリングツールである

*1 <http://taku910.github.io/mecab/>

表 1 実験に用いたデータセット

データ種別	分量
学習データ	正例 19,962
	負例 1,524,155
評価データ	正例 426
	負例 1,574

表 2 ニュース種別とデータ数

種別	データ数	種別	データ数
火災	9,253	爆発	2,803
自動車事故	7,167	航空事故	988
電車事故	4,684	その他事故	982
システム障害	508	社会インフラ	1,076
地震	112	気象	680
自然災害	179	水害	1,012
水難・海難	909	その他気象	982
殺人	296	強盗	637
誘拐	39	通り魔・不審者	902
自殺・変死	600	その他事件	1,823
病気	461	動物	416
その他	4,017		

表 3 評価実験結果

正例	負例	F 値 (最大)	F 値 (平均)
1	1	0.711	0.703
1	10 (W2V)	0.722	0.712
1	10 (RNN)	0.701	0.691
23	1	0.744	0.736
23 (RNN)	1	0.732	0.723
23	10 (W2V)	0.705	0.699
23	10 (RNN)	0.713	0.700
23 (RNN)	10 (RNN)	0.705	0.701

3.2 実験結果

表 3 に実験結果を示す。表の正例、負例の列はそれぞれのクラス数を示す。マルチクラスの場合には、分類手法をクラス数の後ろに記している。(W2V) は Word2Vec による分散表現を用いてクラスタリングしたもの、(RNN) は RNN による分散表現を用いてクラスタリングしたもの、クラス数の後に表記がないものは手で分類したものをそれぞれ表している。F 値 (最大) は、3 回学習したモデルのうち F 値が最大のもの、F 値 (平均) は 3 回学習したモデルの F 値の平均である。

正例、負例ともに 1 クラスの場合と比較し、正例か負例のどちらか一方のみをクラスを分けをすることで多くの場合に性能が向上していることがわかる。最も性能が高かったのは、正例を手動でマルチクラス化し、負例は 1 クラスとした場合で、正例、負例ともに 1 クラスに分類した場合と比較して、F 値 (最大) で 0.033 ポイント、F 値 (平均) でも 0.033 ポイント上回った。正例を自動でマルチクラス化した場合には、手動でマルチクラス化した場合と比べると精度の向上が小さかったが、正例、負例ともに 1 クラスに分類した場合との比較では F 値 (最大) で 0.021 ポイント、F 値 (平均) で 0.020 ポイント性能が向上した。また、正例を 1 クラスとし、負例を Word2Vec の分散表現を用いて自動でマルチクラス化した場合にも、F 値 (最大) で 0.011 ポイント、F 値 (平均) で 0.009 ポイント性能が向上した。一方で、正例、負例をともにマルチクラス化した場合には、性能の向上が見られなかった。

3.3 考察

正例、負例とも 1 クラスに分類する場合と比較し、どちらか一方のみをマルチクラス化した場合にはそれぞれ性能が向上した。表 3 の結果より、正例のみをマルチクラス化する場合、手動でマルチクラス化したほうがよりよい性能が得られるが、自動でマルチクラス化した場合でも性能が向上した。学習データの細分化はタスクに依らず可能であり、また自動で分類した場合でも性能の向上が可能であることから、他のタスクにおいても同様の手法で導入が可能である。性能が向上した理由については、例えば正例であれば「6 丁目で火事らしい。消防車ウーウー言ってる」とい

bayon*2を利用した。負例をマルチクラス化する際のクラス数は、予備実験の結果から最適であった 10 クラスとする。

ニューラルネットワークを用いた tweet 分類手法の実装には Chainer[12] を用いた。RNN の実装は LSTM(Long Short-Term Memory) を利用し、活性化関数には Relu を、学習に用いる誤差計算には softmax cross entropy 法を、パラメータ最適化には Adam[13] をそれぞれ用いた。中間層のノード数は双方向 RNN が 200、Feed-Forward NN の 2 層は入力層に近い方から順に 200、100 とした。正例、負例のマルチクラス化の有無によりモデルの複雑さが変わるため、最適な epoch 数はモデルにより異なることが考えられる。そこで、今回は epoch 数の最大を 10 とし、各 epoch 終了時点でのモデルで性能評価をし、最も精度が高かったものを用いることとする。また、ニューラルネットワークでは学習時にランダム要素が多く、学習ごとに性能が大きく変わるため、本稿では各手法とも同じ条件で 3 回別々に学習し、その中で性能が最も良かったものと、3 回の学習での性能の平均により評価する。学習のミニバッチサイズは 128 とし、マルチタスク学習部分については epoch 数を 3 とした。

評価は、マルチクラス化の有無による条件を合わせるため、ニュースの分野に関わらずにその tweet がニュース制作に役立つものであるかどうかの 2 値で判定した。抽出した tweet のうち、正しくニュースに役立つものである tweet の割合を表す適合率と、本来抽出すべきニュースに役立つ tweet のうちのどれだけの割合のものを抽出できたかを表す再現率の調和平均である F 値により性能を比較した。

*2 <https://github.com/fujimizu/bayon>

う火事に関する情報と、「え、待って電車止まった 踏切のど真ん中で人が立っているとかなんなの」という電車に関する情報など、内容が大きく異なる情報が含まれる。これらを一つの「正例」として扱うより、内容ごとに分けて扱うことでより正確に学習することができたものと考えられる。

一方で、正例、負例ともマルチクラス化した場合には、どちらか一方のみをマルチクラス化した場合と比較して性能が低下した。正例のみをマルチクラス化した場合には正しく判定できたもので、正例と負例をともにマルチクラス化した場合には誤った例として、「(略) 友達が鍋やっつて 火災報知器鳴って起きた (略)」という tweet があった。この例はニュースに役立つ情報ではないため、本来は負例であるが、正例と負例をともにマルチクラス化したモデルでは正例として出力した。この原因として、負例をマルチクラス化した際に「火災報知器」という言葉を含む tweet がいろいろなクラスに分散してしまったことが挙げられる。学習データ中に「火災報知器」を含む tweet は 99 あり、そのうち正例に 58、負例に 41 が分類されているが、負例をマルチクラス化した際に、これらの tweet が複数のクラスに分散した*3。これにより、「火災報知器」を含む負例をよくモデル化できるクラスが存在しなくなったと考えられる。その結果、この tweet は負例の各クラスでそれぞれ低いスコアとなり、相対的に正例のスコアが高くなったものと考えられる。負例のみをマルチクラス化した場合には、正例が雑多な内容をまとめて扱っていたため「火災報知器」に対する正例のスコアも相対的に低く、正しく負例に分類できていたが、正例も分割したことで正例のスコアがより高くなったために誤りになったものと考えられる。これを防ぐためには、マルチクラス化する際のクラスタリングの精度を向上するか、クラス数を減らすなどして、より似た tweet を同じクラスに集まるようにする必要がある。

負例の分類手法については、正例、負例ともマルチクラス化した場合において、Word2Vec を用いた場合と RNN を用いた場合で性能の差はなかった。一方で、負例のみをマルチクラス化した場合には Word2Vec を用いた場合が良好な性能であった。RNN を用いたマルチクラス化では、RNN の学習の際に正例をマルチクラス化したものを用いている。この結果を用いて負例をマルチクラス化した学習データは、正例を同じくマルチクラス化していないと効果が出づらいものと考えられる。Word2Vec による負例のマルチクラス化に用いた tweet の分散表現は wikipedia から学習しており、今回のタスクであるニュース制作に役立つ tweet かどうかを判定するということから独立している。単純に意味の似た tweet をまとめることができたため、負例のクラス化においては有効であったものと考えられる。

*3 Word2Vec を用いた手法の場合で 5 つのクラスにそれぞれ 24, 11, 3, 2, 1 個、RNN を用いた手法では 5 つのクラスにそれぞれ 27, 7, 3, 2, 2 個含まれるように分類された。

4. 関連研究

ソーシャルメディアからの情報抽出の研究は数多く報告されている。その中で、ニュースに役立つ情報に着目した研究としては、Vosecky らは twitter 上の情報の時間的な移り変わりや、話題の中心となる固有名詞に着目した Multi-Faceted Topic Model により、現在話題になっているトピックに関する関連語句を自動抽出したり、高い精度でのクラスタリングが可能であることを示した [14]。また、Hayashi らは NMF をベースとした手法に、スパムなどの自動生成された不要な情報をフィルタリングするための “Hijack Filtering” を組み合わせることでより高い精度での情報抽出を実現している [15]。しかし、これらはいずれも tweet を単語単位で扱う手法である。2.1 節で述べたように、tweet は口語体で書かれることが多く、単語分割が難しい。また、tweet で使われる語彙は非常に数が多いため、全てを扱うモデルを作成するのは困難である。

Tweet を文字単位で扱う手法も数多く報告されている [6], [16], [17]。これにより、単語分割が必要なくなり、口語体の文や twitter 固有の表現なども扱うことが容易になる。また、モデルへの入力文字種の異なり数に減らすことができるため、モデルの学習も容易になる。その一方、アテンションメカニズムなど、近年盛んに研究をされている手法を文字単位での手法に取り入れた研究例は少ない。我々は Dhingra らが提案した Tweet2Vec モデル [6] をベースに、アテンションメカニズムとマルチタスク学習を導入したモデルを用いている。

近年では、tweet の入力部分に RNN ではなく、畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) を用いる手法が多く報告されている。Kim や Guggilla らは、CNN を用いた単語単位のモデル化により、多くのタスクにおいて、LSTM を用いた手法を含めた state-of-the-art の手法を上回ったことを報告している [18], [19]。また、Zhang らは CNN を用いた文字単位でのモデル化手法を提案し良好な性能を挙げている [20]。我々も今後、CNN を用いた手法について検討していく必要がある。

情報抽出の精度を向上するための素性選択の研究も数多くされている。武井らは、その投稿が現実についてのものなのか、ドラマやアニメの内容についてのものなのかを判定するために、tweet にドラマやアニメのタイトルを含むかどうかの素性を追加することで、ニュースに役立つ tweet 抽出の性能を向上した [2]。また、Kanouchi らは tweet で書かれている事象が本人に起きたのか、家族なのか、それ以外なのかなどの特定のために、多くの素性を作成し性能を向上している [21]。本稿の提案手法は tweet の本文のみを扱っているが、将来的にこれらの有効な素性を組み込むことによる性能の向上を検討する必要がある。

5. おわりに

本稿では、ニュース制作に役立つ tweet を自動で抽出する手法について述べた。入力 tweet を、文字ごとに双方向の再帰型ニューラルネットワーク (RNN) に入力することでベクトル化し、その tweet がニュース制作に役立つかを判定する手法を用い、出力をマルチクラス化する場合の性能を比較した。出力 tweet に対し「ニュースに役立つかどうか」の 2 値判定する場合に比べ、正例か負例のどちらか一方を細分化することで性能が向上した。出力のうち正例を手作業によりマルチクラス化した場合が最も良い性能となり、正例か負例かの二値分類をする場合と比較して、F 値が最大値で 0.033 ポイント向上し 0.744 に、また 3 回学習した場合の平均では 0.033 ポイント向上し 0.736 となった。また、正例を自動でマルチクラス化した場合においても性能が向上しており、同じく正例か負例かの二値分類をする場合と比較して、F 値が最大で 0.021 ポイント向上し 0.732 に、平均では 0.020 向上して 0.723 となった。このことから、今回のタスクにおいて、学習データのラベルを細分化することの有効性を確認することができた。一方、正例と負例の双方をマルチクラス化した場合には、正例と負例のどちらか片方のみをマルチクラス化した場合と比較して、性能が低下した。

今後の課題として、正例と負例の双方をマルチクラス化した場合に性能が低下した原因をより詳細に調査する必要がある。また、本稿で述べた手法では tweet の文字情報のみを用いて分類しているが、今後はその他の特徴量を追加するなどして性能の向上を目指す。

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 No11, pp. 290–293 (2014).
- [2] 武井友香, 宮崎太郎, 山田一郎, 後藤淳: “ニュース取材支援のための Tweet 判別手法の検討。” 電子情報通信学会総合大会講演論文集, D-9-40, pp. 130 (2017).
- [3] 鳥海心, 宮崎太郎, 後藤淳, 山田一郎, 八木伸行: “鉄道トラブルに関するツイートの自動抽出手法。” 第 23 回言語処理学会年次大会発表論文集, D4-2, pp. 422–425 (2017).
- [4] 宮崎太郎, 鳥海心, 武井友香, 山田一郎, 後藤淳: “ニュース制作に役立つ 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).
- [5] 萩行 正嗣: “選択式天気情報を用いたソーシャルメディアからの有用投稿抽出。” NLP2016, pp. 397–400 (2016).
- [6] Bhuvan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl and William W. Cohen: “Tweet2Vec: Character-Based Distributed Representations for Social Media.” In Proceedings of ACL2016, pp. 269–274 (2016).
- [7] Dzmitry Bahdanau, Kyunghyun Cho and Yoshua Bengio: “Neural Machine Translation by Jointly Learning to Align and Translate.” ArXiv: 1409.0473 (2014).
- [8] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel and Yoshua Bengio: “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention.” ArXiv: 1502.03044 (2015).
- [9] Munh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals and Lukasz Kaiser: “Multi-task Sequence to Sequence Learning.” ArXiv: 1511.06114, (2015).
- [10] Anders Søgaard and Yoav Goldberg: “Deep multi-task learning with low level tasks supervised at lower layers.” In Proceedings of ACL2016, pp. 231–235 (2016).
- [11] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean: “Efficient Estimation of Word Representations in Vector Space.” ArXiv: 1301.3781, (2013).
- [12] Seiya Tokui, Kenta Oono, Shohei Hido and Justin Clayton: “Chainer: a Next-Generation Open Source Framework for Deep Learning.” In Proceedings of NIPS 2015 workshop (2015)
- [13] Diederik Kingma and Jimmy Ba: “Adam: A Method for Stochastic Optimization.” ArXiv: 1412.6980 (2014).
- [14] Jan Vosecky, Di Jiang, Kenneth Wai-Ting Leung and Wilfred Ng: “Dynamic Multi-Faceted Topic Discovery in Twitter.” In proceedings of International Conference on Information and Knowledge Management (CIKM), pp. 879–884 (2013).
- [15] Kohei Hayashi, Takanori Maehara, Masashi Toyoda and Ken-ichi Kawarabayashi: “Real-Time Top-R Topic Detection on Twitter with Topic Hijack Filtering.” In proceedings of the 21st International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD), pp. 417–426 (2015).
- [16] Soroush Vosoughi, Prashanth Vijayaraghavan and Deb Roy: “Tweet2Vec: Learning Tweet Embeddings using Character-level CNN-LSTM Encoder-Decoder.” In proceedings of the 39th International conference on Research and Development in Information Retrieval (ACM SIGIR), pp.1041–1044 (2016).
- [17] Svitlana Vakulenko, Lyndon Nixon and Mihai Lupo: “Character-based Neural Embedding for Tweet Clustering.” In proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, pp. 36–44 (2017).
- [18] Yoon Kim: “Convolutional Neural Networks for Sentence Classification.” In proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746–1751 (2014).
- [19] Chinnappa Guggilla, Tristan Miller and Iryna Gurevych: “CNN- and LSTM-based Claim Classification in Online User Comments.” In proceedings of the 26th International Conference on Computational Linguistics (COLING), pp. 2740–2751 (2016).
- [20] Xiang Zhang, Junbo Zhao and Yann LeCun: “Text Understanding from Scratch.” ArXiv 1502.01710 (2015).
- [21] Shin Kanouchi, Mamoru Komachi, Naoaki Okazaki, Eiji Aramaki and Hiroshi Ishikawa: “Who caught a clod? – Identifying the subject of a symptom.” In proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1660–1670 (2015).