

Twitter 上でトレンドの言葉を含む投稿に対する ニュース価値の分類手法の提案

M2016SC012 岡本裕貴

指導教員：河野浩之

1 はじめに

近年、SNS(Social Networking Service)の普及により、誰でも好きな話題について SNS 上に投稿することができるようになった。その中でも Twitter で発言を投稿するには、パソコンやスマートフォンで自身のアカウントにログインし、140 文字以内で内容を入力し「ツイート」ボタンを押すことで投稿することができる。現在では Twitter は(2018 年 10 月)世界で約 3 億 2600 万人のユーザーが使用している^{*1}。ツイートは電子メールの送信や、SMS テキストメッセージの送信など、幅広い Web ベースのサービスを使用してスマートフォンから直接送信することができる。しかしながら、Twitter 上には様々な投稿がありふれており、ユーザーが投稿した記事がニュース価値のあるツイートかどうかを見極めるのは困難である。本研究では、Twitter 上で投稿されたトレンドのキーワードを含む記事に対して、ニュース価値のある記事であるかどうかの分類を行うことを目的とする。実験方法は、Twitter REST API を用いてデータを収集し、Excel を用いてデータセットの作成と保存を行う。また、ニューラルネットワーク(NN)の実装を行うことができる Chainer を用いて、ニュース価値のある記事の分類システムを Python で実装を行う。そして、提案モデルと SVM や J48 との性能評価を Weka を用いて比較を行う。評価の指標として Accuracy(正解率)、False Positive Rate(偽陽性率)を用いる。

本研究の構成を以下に示す。2 章では、ニュース制作に役立つツイートを分類する研究や、ニュース性のあるツイートが既出かどうか分類する研究についての紹介と問題点を提起する。3 章では、Twitter で投稿された記事がニュース価値があるかどうかを分類する手法を提案する。4 章では、Twitter REST API を用いてデータセットの作成と、Chainer を用いてニューラルネットワークの提案モデルの実装を行う。5 章では、本研究の提案手法の性能評価をし、6 章はむすびを記述する。

2 ニュース価値のあるツイートを分類する先行研究

本章では、2.1 節にニュース制作に役立つツイートを分類する研究の紹介、2.2 節に Twitter 上でニュース性のある Tweet に対して、その Tweet が既出であるかどうか分類する研究について紹介し、2.3 節に問題点と改善点を挙

げる。

2.1 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 [1]

宮崎等は、ニュースに役立つツイートを分野ごとに細分化して学習し、そのツイートがニュース制作に役立つかを分類する手法を提案した。そして、入力ツイートを文字ごとに双方向の RNN に入力することでベクトル化して、そのツイートがニュースに役立つか判定を行い、出力をマルチクラス化する場合の性能を比較した。その結果、手作業でニュースに役立つツイートを分類した場合、情報抽出の精度が F 値で 0.744 となり、自動で分類した場合は 0.732 という結果が得られた。

2.2 ニューラルネットワークを用いた既出 Tweet 分類 [2]

牧野等は、ニュース性のある Tweet が既出であるか、非既出であるかどうかの分類を行った。報道機関やまとめサイトの Tweet およびその引用などの既出の情報を用いた Tweet を既出 Tweet とし、第一報となりうる Tweet を非既出 Tweet と定義した。そして、ニュース性のある Tweet を抽出した結果に対する後段の処理として、抽出した Tweet の本文を用いて既出 Tweet かどうかを NN ベースの手法を複数用いて分類するシステムの提案を行った。その結果、分類処理として CNN の中間層出力を結合し、FFNN で出力する処理を 5 epoch で 37 分程度学習させることで、約 88 % という F 値で分類することができた。

2.3 先行研究の問題点と改善点

宮崎等の研究では、ツイートの文字情報のみを用いて分類を行っているため、そのほかの特徴量を追加することで性能の向上を目指すことができる。牧野等は、引用文が文中に入っているツイートに対して、ニュース性のある既出ツイートかどうかの判定をしている。しかし、文中に引用文が入っていても、ニュース価値のあるツイートも多く存在するので、URL を含むツイートだけでなく、ユーザーの意見を取り入れたツイートを取得して分類を行う。

3 ニュース価値のある記事の分類手法の提案

3 章では、2 章で紹介した先行研究の問題点と解決方法を踏まえて、ニュース価値のある記事の分類手法の提案を行う。3.1 節では提案手法の流れを説明し、3.2 節では、実装するアルゴリズムの紹介をする。そして、3.3 節ではツイート収集をする方法について紹介する。

^{*1} <http://investor.twitterinc.com/home/default.aspx>

3.1 ニュース価値分類システムの構成図

本節では、3.1 節の問題点と提案を考慮し、ニュース価値分類システムを構成するために必要なアーキテクチャを図 1 に示す。

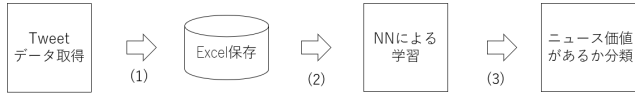


図 1 ニュース価値のある記事を分類する流れ

- (1) では Twitter のデータの収集を行う。Twitter REST API を使用して、トレンドとなっているキーワードを指定して 1000 件の Tweet を取得し、Excel に保存する。本研究では 9 月 30 日から 10 月 1 日にかけて日本に上陸した「台風」と、1 月 18 日にトレンドとなった「センター試験」というキーワードに関するデータの収集を行った。
- (2) では Excel を用いて保存したデータの前処理を行う。そして、形態素解析のツールである KHCoder を用いて、それぞれのキーワードに対してニュース価値のありそうな言葉を抽出した。「台風」では、「影響」、「塩害」、「被害」の 3 種類を選び、「センター試験」では、「受験」、「数字」、「studyplus」の 3 種類の言葉を選択した。次に、学習データとテストデータを 4:1 に分割する。そして、提案する NN の学習モデルを用いて実装を行う。
- (3) では (2) の訓練データを用いて学習したモデルを使用して、テストデータがニュース価値のある記事かどうかの分類を行う。

3.2 提案モデルの式

本研究の提案モデルはニューラルネットワーク (Neural Network) を用いる。NN は人間や動物の脳の神経細胞の作り出すネットワーク構造を模して作られたアルゴリズムの一つである。また、既存の方法よりも優れており、自然言語処理の分野で高い成果をあげているアルゴリズムの一つである。図 2 に一般的な NN を展開したモデルを示す。(1) は入力層から隠れ層への流れる計算式を示す。それぞれの線に重み w とバイアス b が設定されている。次に活性化関数 $f(x)$ を通して (2) のような出力を得る。そして、隠れ層のユニット数や層の数を増やして同様の計算を順に行う。本研究では活性化関数として、ReLU 関数 (3) を用いる。ReLU 関数は入力した値が 0 以下のとき 0 になり、1 より大きいとき入力をそのまま出力する。深層学習では、モデルのパラメータを学習するために誤差を微分し、学習を進めることができる。

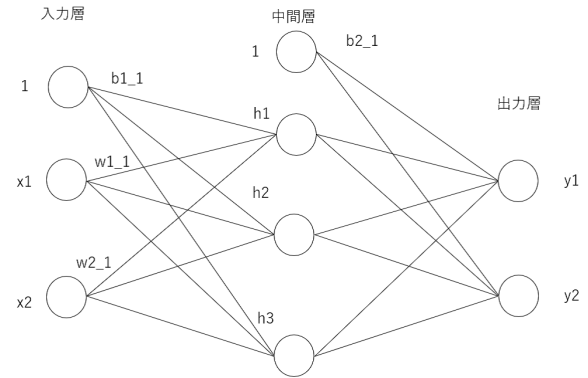


図 2 ニューラルネットワークの概念図

$$u_1 = x_1w_{11} + x_2w_{12} + x_3w_{13} + b_1 \quad (1)$$

$$h_i = f(s_i) \quad (2)$$

$$f(x) = \max(0, x) \quad (3)$$

3.3 Tweet 収集ツール

Twitter 上でユーザーの投稿の中から、トレンドとなっているキーワードを含む記事のデータ収集を行う。データの収集には Twitter API を使用し、Python3.6 で実装を行う。Twitter API は Twitter, Inc. が提供しているサービスであり、Web サイトやアプリなどから Twitter の機能呼び出すことができ、この API を利用することでツイートの参照や検索などを行えるアプリケーション開発ができる。Twitter API にアクセスするには Twitter Application Management でアプリケーションの登録と 4 つの認証キーの取得が必要である。本研究では、ツイートデータ取得のための条件指定がしやすく、データの取得に時間がかからないので、最も適している REST API を用いる。また、収集したツイートデータを保存するために、Excel を用いてデータベースに保存する。そして、ユーザーが Tweet した文章を品詞情報を持った単語に区切るために、形態素解析のツールとして KH Coder を用いる。

4 Chainer によるニュース価値の評価実験

本章では、3 章で紹介したツールを用いて、ニュース価値の分類の実験を行う。4.1 節では実験環境を示し、4.2 節では収集したデータについて説明を行う。そして、4.3 節では NN のパラメータの設定について記述する。

4.1 実験環境

本研究の実験環境は、OS: Windows10, メモリ 8GB, Intel(R) Core(TM) i7-4790 CPU の環境下で行う。システムの実装には、Python パッケージを一括でインストール可能にしたパッケージである Anaconda3-4.4.0 で Python3.6 を使用。Python で Twitter API を操作する

ために、Tweepy3.5.0 というライブラリを用いる。そして、ニューラルネットワークを実装するために Chainer4.0 を用いる。Tweepy や Chainer のインストールには Anaconda Prompt を開いて、インストールコマンド (pip) を行った。その他に、Chainer の実装環境を整えるために、マイクロソフトが提供するプログラム作成のためのツールである、Visual Studio 2015, NVIDIA が提供する GPU 向けの C/C++ 言語の統合開発環境 CUDA8. 0 や Deep Learning 用のライブラリ cuDNN をインストールすることで、Chainer を実装する。また、プログラムの記述の際に使用するエディタとして Visual Studio Code を用いた。

4.2 データ収集

データの取得には Twitter REST API を用いる。REST API を使用することで、プログラム上から Twitter のデータにアクセスして読み込みと書き込みを行うことができる。

表 1 に、収集したデータについて示す。収集した一つ目のデータは 1000 件であり、9 月 30 日にツイートされた「台風」というキーワードを含む投稿を Twitter 上で集めた。二つ目は 1 月 18 日に 582 件の「センター試験」というキーワードを含む投稿を集めた。また、データを収集する際にツイートをしたユーザーの特徴を得るために、ユーザ名、場所、フォロー数、フォロワー数、ツイート数、ID、投稿時間、リツイート数、お気に入り数、テキスト内容の 10 種の属性を同時に取得し、Excel に保存した。

表 1 データ収集

データ収集日	2018 年 9 月 30 日, 2019 年 1 月 18 日
データ件数	1000 件 (台風), 582 件 (センター試験)
属性	ユーザ名, 場所, フォロー数, フォロワー数, ツweet数, ID, 投稿時間, リツイート数, お気に入り数, テキスト内容

次に、収集したデータの中で前処理を行った属性を示す。

- ユーザー名: Twitter のユーザー名は曖昧な表現で表されたものが多い。よって本研究では、絵文字や顔文字で構成される名前を 0 とし、日本語で表記されたものは 1, 英語で表記されたものは 2 として 3 種類に分けた。
- 場所: 位置情報を付加したツイートと固定した場所でのツイートが見られた。場所が書かれていないツイートを 0, 場所が不明のツイートを 1, 日本国内のツイートを 2 とした。
- テキスト内容: 検索したキーワードを含む記事の中から、KHCoder を使用して形態素解析を行い、出現回数の多い上位 20 種類の言葉を含む記事の中から「台風」と「センター試験」のそれぞれに対して、関係性の深い言葉を 3 種類選択し、その言葉を含む記事がニュー

ス価値のある投稿として正解ラベルを割り当てた。

4.3 Chainer を用いた NN のパラメータ設定

Chainer を用いてニューラルネットワーク (NN) のパラメータの設定を行う。エポック数の設定では、エポック数を大きくすると学習の回数が増える。ミニバッチサイズは、学習データをいくつかのサイズに分割しておき、その分割された少量のサンプルを用いてニューラルネットワークのパラメータを更新していく。ミニバッチサイズを大きくすると学習が早くなるが、大きくしすぎるとパラメータが最適解に収束しにくくなる傾向があるので、学習過程の誤差値を見ながら設定を行った。本研究では、エポック数は 100, ミニバッチサイズは 8 に設定して実験を行った。次に、ニューラルネットワークの登録を行う。本研究では、誤差の計算に softmax_cross_entropy という損失関数を使うことを設定してモデルの作成を行う。損失関数はニューラルネットワークの出力と学習データの差を計算するための関数である。そして、学習データと計算した結果の誤差を決めるために最適化関数の設定を行う。本研究では、最適化関数として Adam (Adaptive moment estimation) を用いる。これは、ほかの最適化関数である AdaGrad や RMSProp, AdaDelta を改良したものであり、現状最も評価されているので Adam を用いる。

5 Chainer による分類システムの性能評価

本章では、4 章で記述したニューラルネットワークを用いたニュース価値の分類結果を示す。収集したデータが正しく分類されているかを評価するために、accuracy (正確度), precision (適合率) を用いる。5.1 節では、収集したデータに対する分類結果を示す。5.2 節では、既存の方法である SVM や J48 との性能比較を示す。

5.1 ニュース価値のある記事の分類結果

本研究で実装したニューラルネットワークによるニュース価値の分類結果を示す。

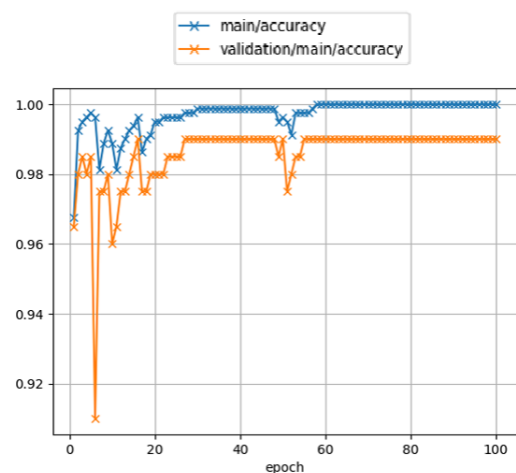


図 3 正解率 (台風)

図 3, 図 4 にそれぞれ「台風」, 「センター試験」のキーワードで取得し, ニュース価値のある記事かどうかを分類した結果を示す。「台風」におけるニュース価値の分類では, 学習回数が 30 回でテストデータの分類精度が 1.0 になり正しい分類ができたことが分かった。また, 学習用データも 0.99 に収束し, かなり精度の高い分類ができた。

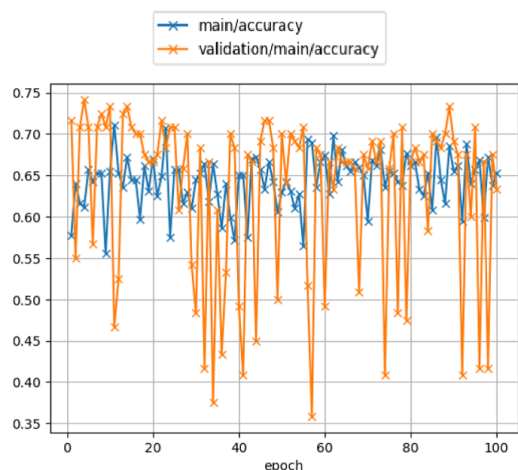


図 4 正解率 (センター試験)

「センター試験」におけるニュース価値の分類では, テストデータは学習回数が 23 回で最も精度がよくなり, 0.71 の精度の分類ができた。学習データでは, 学習回数が 4 回目で最も良い分類ができ, 0.74 の分類ができた。

これらの結果から「台風」によるキーワードでの分類精度が高かったのは, 「台風」というキーワードを含むツイートを投稿したユーザーのリツイート (引用ツイート) による投稿が多かったからと考えられる。また, 正解ラベルとして選択した 3 種類の言葉を含む投稿が, 台風による被害や影響を心配するツイートが集中したことや, 不確かな情報のツイートや嘘のツイートを発するユーザーがいなかったため, 学習したデータの種類の少なくなり分類精度が高くなったと考えられる。

5.2 提案手法との性能評価の比較

SVM と J48 は Weka を用いて実行し, 5.1 節と同様に「台風」と「センター試験」のキーワードにおけるニュース価値の分類精度を表 2 に, 正解との誤差を表 3 に示す。なお, 提案手法では学習回数が「台風」では, 24 回目, 「センター試験」では, 99 回目の時に最も良い精度が出たので, これを基に比較を行う。

SVM と J48 による「台風」のニュース価値の分類では, どちらも分類精度が 1.0 に近く, FPRate も 0 に近いので, 正しい分類ができた。そして, 「センター試験」のニュース価値の分類では, SVM が最も正しい分類ができた。しかし, SVM と J48 のどちらも FPRate が高く, 実際に間違っ

表 2 性能評価比較 (精度)

キーワード	「台風」	「センター試験」
指標	Accuracy	Accuracy
SVM	0.998	0.816
J48	0.979	0.704
NN(提案手法)	1.000	0.731

表 3 性能評価比較 (誤差)

キーワード	「台風」	「センター試験」
指標	FPRate	FPRate
SVM	0.005	0.500
J48	0.049	1.000
NN(提案手法)	0.010	0.260

い分類ができたが, 「センター試験」では SVM のほうが 8.5 % 程上回る精度が見られた。しかし, FPRate では提案手法の方が, 24 % 程良い精度を出すことができた。この結果から, 「センター試験」での分類結果は, 提案手法よりも SVM が 8.5 % 程良い精度を出したが, FPRate は提案手法の方が 24 % も低いので, 提案手法の NN の分類精度と FPRate が最もバランスの良い分類ができていると考えられる。

6 むすび

本研究では, Twitter 上でトレンドとなっているキーワードを含む記事が, ニュース価値があるかどうかの分類を目的として研究を行った。提案手法では, Chainer を用いてニューラルネットワークの実装を行い, 既存の方法と性能評価を行った。SVM と J48 と比較を行ったところ, 提案手法よりも SVM が 8.5 % 程 Accuracy が高かったが, FPRate は提案手法の方が SVM よりも 24 % 程良い結果だったので, 既存の手法よりも間違っ

参考文献

- [1] 宮崎 太郎, 鳥海 心, 武井 友香, 山田 一郎, 後藤 敦, " 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", 情報処理学会研究報告.Vol.2017-IFAT-127 No.1, pp1-6, 2017.
- [2] 牧野 仁宣, 武井 友香, 宮崎 太郎, 後藤 敦, " ニューラルネットワークを用いた既出 Tweet 分類", 人工知能学会全国大会論文集 29, pp.1-4, 2015.
- [3] 新納 浩幸, 「Chainerv2 による実践深層学習」, オーム社, 2017.
- [4] 牧野 浩二, 西崎 博光, 「Python による深層強化学習入門」, オーム社, 2018.