# ニューラルネットワークを用いた既出 Tweet 分類

牧野 仁宣　　武井 友香　　宮﨑 太郎　　後藤 淳

NHK 放送技術研究所 〒157-8510 東京都世田谷区砧 1-10-11

E-mail: {makino.k-gg, takei.y-ek, miyazaki.t-jw, goto.j-fw}@nhk.or.jp

## 1. はじめに

近年，Twitter や Facebook などの即時性が高い SNS の投稿から番組素材となる情報を抽出し，ニュースや番組の制作に活用するケースが増えている[1]．しかし，人手による情報抽出は，大変な労力がかかる．こうした理由からニュース性のある投稿を自動で抽出する研究[2]がなされてきた．またNHKでも，実際のニュース現場における抽出Tweetを基に機械学習させ，様々な話題が含まれる Tweet の中からニュース性のある Tweet を抽出するシステムの研究・開発を進めている．これまでに，情報を整理して掲示するため，抽出した Tweet を自動分類して表示するシステム[3]や，位置情報の含まれる Tweet を地図上に掲示するシステム[4]を開発している．しかし，Twitter で発信されるニュース性のある情報は多種多様でかつ多量であるため，掲示される全ての Tweet を監視することは依然として困難である．加えて，その情報の活用方法は，利用者・用途・状況に応じて大きく異なる．こうした理由から，それぞれの環境で必要とされる Tweet に限った掲示が求められる．

例えば，報道現場で利用可能なニュース性のある Tweet は，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（以下，既出 Tweet），引用ではなく，第一報となりうる Tweet（以下，非既出 Tweet）に分類できる．情報を抽出する目的が、ニュース素材として必須の条件である第一報取得の場合、非既出 Tweet を収集することが重要となる．一方、ニュースに対しての意見収集や続報の取得が目的である場合には、既出 Tweet も収集する．このように，利用目的に応じて，情報の使い分けが必要となる．

既出 Tweet は二種類に分類することが可能である．引用元の出典が含まれていればキーワードフィルタリングで抽出が可能となる(以下，出典あり既出 Tweet)が，含まれていなければ(以下，出典なし既出 Tweet)単純な処理では既出であるとの分類は難しい．

こうした理由から，単純なキーワードフィルタリングではなく，様々な文の条件を考慮して分類する処理を用意することが必要である．そこで本稿では，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 か否かを機械学習により分類するシステムを提案する．分類処理手法として，RNN (Recurrent Neural Networks)，CNN (Convolutional Neural Networks)等のニューラルネットワーク (NN: Neural Networks)を用いる手法[5-8]が考えられる．本稿ではこれらの様々な NN 構成に対し，複数の系列を入力し，組み合わせる手法等を比較検討し，特性を評価したため報告する．

## 2. 既出 Tweet 分類システム概要

既出 Tweet 分類システムの全体構成を図 1 に示す．まず，「ニュース性 Tweet 抽出処理」で，全ての日本語 Tweet の 10%ランダムサンプルを入力し，抽出システムでニュース性のある Tweet を抽出し，残りのニュース性が無いと判断された Tweet は破棄する．続いて「既出・非既出分類処理」で，抽出されたニュース性のある Tweet を，既出 Tweet と非既出 Tweet に分類する．

ニュース性 Tweet 抽出[3]は，全くニュースと関係ない内容が多くを占める Tweet から，0.1%にあたるニュース性のある Tweet を抽出する．ニュースと関係ない Tweet は多種多様であるため，その中から NN を用いて抽出するには膨大なデータから学習する必要がある．一回の処理で抽出に加えて既出・非既出の分類をするには，抽出過程で廃棄される，分類する必要のない Tweet も含めて分類を学習することになる．また，既出・非既出以外の分類への拡張など，分類構成を変更する度に再度全体のシステムを構築し，抽出の大規模な学習も含めて再学習をする必要が生じる．ユーザの要求に合わせた分類掲示を増やす度に，こうした再学習をすることは現実的ではない．こうした理由から，本稿では抽出処理と分類処理に分けた二段階構成システムを提案した．

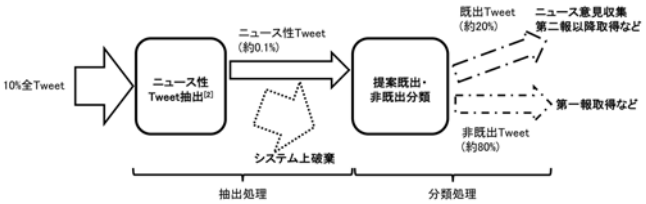


**図 1: 既出 Tweet 分類システム構成**

### 2.1. ニュース性 Tweet 抽出処理

本稿のニュース性 Tweet 抽出には既に提案済の手法[3]を用いた．Tweet を文字毎に one-hot ベクトル系列化した後，アテンション機構を用いた Bi-directional LSTM に入力し，ニュース性の無い Tweet と，ニュース性のある 24 のマルチクラス Tweet に分類する．このマルチクラスは，既出・非既出とは全く相関の無い独立な分類である．ここで用いるモデルは，実際に報道現場で抽出された Tweet を基に，教師あり学習により学習した．

### 2.2. 既出・非既出分類処理

既出・非既出分類処理は，学習データを基に様々な分類手法を用いて，既出・非既出に分類する処理である．なお，既出・非既出は細かく以下の三種類に分類した．

① 出典あり既出 Tweet
出典が文中にある，報道機関やまとめサイトの Tweet および，その引用を用いた Tweet
e.g.「小型機墜落　4 人は墜落の衝撃で死亡か ｜NHK ニュース　機首から突っ込んで壊れたのかな」

② 出典なし既出 Tweet
出典が文中にない，報道機関やまとめサイトの Tweet および，その引用を用いた Tweet
e.g.「小型機墜落　4 人は墜落の衝撃で死亡か　機首から突っ込んで壊れたのかな」

③ 非既出 Tweet
上記2つ以外の全 Tweet

①, ②はどちらも既出 Tweet であるため, 合わせて正例, ③は非既出 Tweet であるため負例と分類する処理とする.

分類処理への入力は抽出処理の出力に限定されるため, 入力系列の種類が限られる.そのため, 抽出処理ほどの膨大な学習データは必要としない.学習量や学習構造に起因するハードウェア負担と, 高度な処理による特性向上はトレードオフの関係にあるが, 膨大な学習は必要とされないため, 重い処理も可能である. 本稿では, 実験によりこの処理に対する特性を評価する.

## 3. NN を用いた既出・非既出分類手法

既出・非既出分類処理の手法として, NN を用いた手法を複数構成検討する. 入力系列内の時系列を考慮せず FFNN (Feed Forward Neural Networks)のみを用いる手法と, 時系列を考慮した RNN の一種である LSTM (Long short-term memory)[5]にアテンション機構を導入した手法, およびフィルタによりフィルタサイズ内の時系列を考慮する CNN[6]を用いた手法を使用する.

テキストを用いた解析では, 一般的に文字または単語の系列に分割し, その系列をベクトル化したものが入力として用いられる. 本稿では, NN 各手法に対して, 入力系列として文字系列のみを入力とする方法, 単語系列のみを入力とする方法に加え, 3.4 節で述べるこの 2 つの系列を別々に入力し結合する提案法の三種類の入力を用いる.

テキストを文字系列に変換する際は一文字ずつ切り分け, 単語系列に変換する際は辞書として mecab-ipadic-NEologd[9]を用いた, 形態素解析器 MeCab を用いる. 各文字/単語系列は, 以下の二種類ベクトルを NN 構造に応じて用いる.

Ⅰ. one-hot 系列ベクトル
Tweet をそれぞれ文字/単語に分割し, one-hot ベクトル化した後に系列ラベル化. LSTM, CNN で利用.

Ⅱ. BOW (Bag of Words)ベクトル
Ⅰの系列ベクトルを一つのベクトル化. FFNN で利用.

Ⅰ, Ⅱ共に出現頻度を考慮しないベクトルを用いる.

### 3.1. 文字入力 NN/単語入力 NN

文字系列, または単語系列のみを入力として用いる場合の各 NN 手法の構成を以下に示す.

#### 3.1.1. FFNN 手法

図 2 に文字/単語 BOW 入力で FFNN のみを用いる構成を示す. まず入力ベクトル$\boldsymbol{x}$ を入力層 FFNN ($\boldsymbol{W}^{\mathrm{in}}$, $\boldsymbol{b}^{\mathrm{in}}$)に通し, ベクトル$\boldsymbol{h}^{\mathrm{in}}$に出力する. 続いて, 別の FFNN で構成される中間層($\boldsymbol{W}^{\mathrm{int}}$, $\boldsymbol{b}^{\mathrm{int}}$)に通し$\boldsymbol{h}^{\mathrm{int}}$に出力し, さらに出力層 FFNN ($\boldsymbol{W}^{\mathrm{out}}$, $\boldsymbol{b}^{\mathrm{out}}$)に通して 2 次元ベクトル$\boldsymbol{h}^{\mathrm{out}}$に出力し, Softmax 関数で最終出力とする. 学習の際は, Softmax 関数の出力と正解の one-hot ベクトルの間で, 交差関数により誤差を求め, 逆伝播により学習する. 判定の際はこの Softmax 出力の argmax を取ることで出力とする. $\boldsymbol{n}$次元の入力 BOW ベクトルを$\boldsymbol{x}$とすると, 式(1)に従う. なお, 各層間の活性化関数$\boldsymbol{a}(\boldsymbol{x})$は RELU (Rectified Linear Unit)を用いる.

$$\begin{aligned}\boldsymbol{h}^{\mathrm{in}} &= a\left(\boldsymbol{W}^{\mathrm{in}}\boldsymbol{x}+\boldsymbol{b}^{\mathrm{in}}\right)\\ \boldsymbol{h}^{\mathrm{int}} &= a\left(\boldsymbol{W}^{\mathrm{int}}\boldsymbol{h}^{\mathrm{in}}+\boldsymbol{b}^{\mathrm{int}}\right)\\ output &= softmax\left(\boldsymbol{W}^{\mathrm{out}}\boldsymbol{h}^{\mathrm{int}}+\boldsymbol{b}^{\mathrm{out}}\right)\end{aligned} \quad (1)$$

#### 3.1.2. LSTM 手法

図 3 に文字/単語 one-hot 系列ベクトルを入力とし, 中間層としてアテンション機構を導入した LSTM を用いる構成を示す. まず系列ベクトル$\{\boldsymbol{x}\}$ = { $\boldsymbol{x}_0$, $\boldsymbol{x}_1$,… }をそれぞれ入力層 FFNN に通しベクトル系列$\{\boldsymbol{h}^{\mathrm{in}}\}$ =$\{\boldsymbol{h}_0^{\mathrm{in}}, \boldsymbol{h}_1^{\mathrm{in}},\cdots\}$を出力する. その後中間層である LSTM に通し, 単一ベクトル$\boldsymbol{h}^{\mathrm{int}}$を出力する. この時, LSTM の処理にはアテンション機構を導入した構造[5]を用いた. LSTM の構造により, 入力の全時系列を学習する. 出力層以降の処理は FFNN 手法と同様である. 式(2)に従う.

$$\begin{aligned}\boldsymbol{h}_t^{\mathrm{in}} &= a\left(\boldsymbol{W}^{\mathrm{in}}\boldsymbol{x}_t+\boldsymbol{b}^{\mathrm{in}}\right) \quad (t=0,1,\cdots)\\ \boldsymbol{h}^{\mathrm{LSTM}} &= LSTM\left(\{\boldsymbol{h}_t^{\mathrm{in}}\}\right)\\ \boldsymbol{h}^{\mathrm{int}} &= a\left(\boldsymbol{W}^{\mathrm{int}}\boldsymbol{h}^{\mathrm{LSTM}}+\boldsymbol{b}^{\mathrm{int}}\right)\\ output &= softmax\left(\boldsymbol{W}^{\mathrm{out}}\boldsymbol{h}^{\mathrm{int}}+\boldsymbol{b}^{\mathrm{out}}\right)\end{aligned} \quad (2)$$

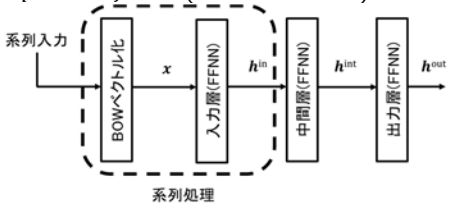


**図 2: 文字/単語各入力 FFNN 構成**

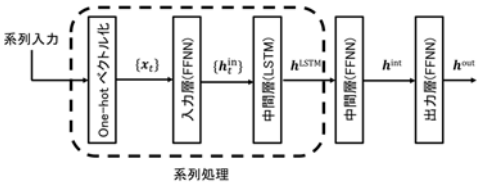


**図 3: 文字/単語各入力 LSTM 構成**

#### 3.1.3. 畳み込みニューラルネット (CNN)手法

図 4 に文字/単語 one-hot 系列ベクトルを入力とし, 中間層として CNN を用いる構成[6]を示す. まず LSTM と同様の処理によりベクトル系列$\{\boldsymbol{h}^{\mathrm{in}}\}$を出力する. その後, $l$ 種類の畳み込み層に入力し, それぞれ出力を max-pooling する. この時, 畳み込み層によりフィルタサイズに応じた時系列を学習する. その全ての出力を結合し, 中間層 FFNN に入力する. 出力層以降の処理は FFNN 手法と同様である. 式(3)に従う.

$$\begin{aligned}\left\{\boldsymbol{h}_t^{\mathrm{CNN},p}\right\} &= a\left(CNN_p\left(\{\boldsymbol{h}_t^{\mathrm{in}}\}\right)\right) \quad (p=0,1,\cdots l-1)\\ \boldsymbol{h}^{\mathrm{pool},p} &= \max_t \boldsymbol{h}_t^{\mathrm{CNN},p}\\ \boldsymbol{h}^{\mathrm{int}} &= a\left(\boldsymbol{W}^{\mathrm{int}}\begin{bmatrix}\boldsymbol{h}^{\mathrm{pool},0}\\ \vdots\\ \boldsymbol{h}^{\mathrm{pool},l-1}\end{bmatrix}+\boldsymbol{b}^{\mathrm{int}}\right)\\ output &= softmax\left(\boldsymbol{W}^{\mathrm{out}}\boldsymbol{h}^{\mathrm{int}}+\boldsymbol{b}^{\mathrm{out}}\right)\end{aligned} \quad (3)$$

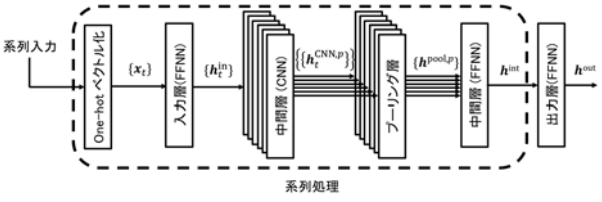


**図 4: 文字/単語各入力 CNN 構成**

### 3.2. 文字単語結合 NN (提案手法)

本研究では, 文字系列・単語系列全体を双方用い, 組み合わせる方式を提案する. 3.1 節では, 文字・単語のどちらかの系列を各 NN 構成に入力する手法を用いていた. しかし, 文字系列は未知語が少ないという利点, 高次元の入力を用い

なくても，入力文章を不足無く表現するという利点，および日本語等の単語間に空白を空けない言語を利用する場合に，単語分割する形態素解析器が必要なく，性能に左右しないという大きな利点がある．しかし，文章は単語系列として書かれるものであり，文字は一要素のみを取り出すと意味を持たないが，単語は一要素のみで意味を持つという利点があるため，双方の利点を活かすことが見込まれる本 NN を提案する．

図 5 に単語と文字のベクトルを中間層で結合する NN 構成を示す．各 NN は，それぞれ文字系列・単語系列から 3.1 節の各手法の系列処理出力ベクトル$\boldsymbol{h}^{\mathrm{NN}}_{\mathrm{word}}$・$\boldsymbol{h}^{\mathrm{NN}}_{\mathrm{char}}$に出力した後に，出力を結合する．なおこの処理は，各手法の図 2-4 の点線部の系列処理部にあたる．その後結合後中間層 FFNN に入力する．以降の処理は文字/単語系列入力 NN の各構成と同様である．一連の処理は式(4)に従う．

$$\boldsymbol{h}^{\mathrm{int}} = a\left(\boldsymbol{W}^{\mathrm{int}}\begin{bmatrix}\boldsymbol{h}^{\mathrm{NN}}_{\mathrm{word}}\\ \boldsymbol{h}^{\mathrm{NN}}_{\mathrm{char}}\end{bmatrix} + \boldsymbol{b}^{\mathrm{int}}\right) \tag{4}$$

$$output = softmax\left(\boldsymbol{W}^{\mathrm{out}}\boldsymbol{h}^{\mathrm{int}} + \boldsymbol{b}^{\mathrm{out}}\right)$$

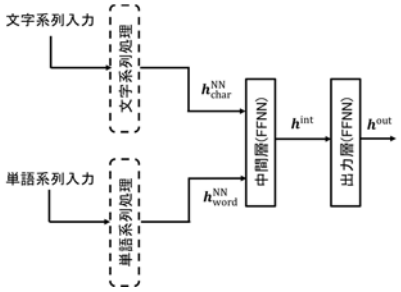


**図 5: 文字単語結合 NN 構成**

## 4. 評価実験

提案分類手法の特性を，実験により評価する．この時，比較手法には，ベースラインとして学習データ中の出典あり既出 Tweet における，明記されている出典をアノテートし，その出典をキーワードとして本文をフィルタリングするキーワードフィルタリング手法を用いた．なお，2.2.の①e.g.では出典は「NHK ニュース」となる．

### 4.1. 実験条件

学習データ・テストデータ共に，抽出処理[3]でニュース性があると判定された Tweet を用いる．アノテータにより①出典あり既出，②出典なし既出，③非既出の三種類に分類した．学習データは 2017 年 6 月 6, 8, 10, 12 日の全 44,670 個の出力 Tweet を用い，テストデータは 2017 年 7 月 6, 8, 10, 12 日からランダムサンプルした 10,000 個の Tweet を用いる．それぞれアノテートした 3 種類の個数を表 1 に示す．

本稿では，深層学習フレームワークとして chainer を，勾配降下法のアルゴリズムとして Adam を用いる．学習回数は 5 回，各中間層出力の次元は 200，ミニバッチサイズ 100 とした．この時，文字，単語共に学習データに 10 回以下しか出現しない要素は未知語として扱うこととする．学習時は各段のノードにドロップ率 0.5 のドロップアウト処理を組み込む．CNN 手法では，(2, 2, 3, 3, 4, 4)のフィルタサイズを持つ 6 種のフィルタを用いる．また，NN はランダムで設定される初期値に特性が影響されるため，20 回同じ実験を独立に試行し，Precision, Recall, F 値の平均を求める．

**表 1: データセット条件**

| | ①出典あり | ②出典なし | ③非既出 |
|---|---|---|---|
| 学習 | 4,273 | 5,027 | 35,370 |
| テスト | 844 | 1,184 | 7,972 |

### 4.2. 実験結果

各入力系列，各分類手法に対して，テストデータの全データを用いた場合の Precision, Recall, F 値結果を表 2 に示す．また，出典のあり/なしに対するモデルの性能差を評価する．正例に対する精度のみを評価するため，テストデータとしてそれぞれ①のみ，②のみ，①+②を用いた場合の Recall のみを表 3 に示す．加えて，各学習に必要とされる時間を表 4 に示す．表 2 より，ベースライン手法は Precision の 94.1%と高い性能が得られているのに対し，Recall は 34.0%と低い特性が得られた．また表 3 より，①と比べて②の Recall は 70%程度低い特性が得られた．

全ての提案手法において F 値がベースライン手法を上回った．F 値が最低値を示す単語入力の FFNN 手法でも，Precision はキーワードフィルタリングのベースラインから 0.6%程度劣化するものの，Recall は倍以上の 79%程度，F 値では 85%程度得られた．また①および②のテストデータによる Recall 差は，23%程度であった．

文字/単語各入力 NN では F 値はどの手法でも文字入力が 1.0%程度高い，また，どちらの入力でも LSTM 手法の F 値は，FFNN 手法と比べ 2.0%程度，CNN 手法と比べ 1.0%程度高い性能であった．文字/単語各入力の最高値を示す手法は，文字入力を用いた LSTM 手法であり，87%程度であった．この時，①および②の Recall 差は 15%程度であった．

文字単語結合 NN では，どの手法も文字入力よりも高い性能が得られた．特に CNN 手法では 2.0%程度の改善が見られ，LSTM 手法とほぼ同程度の 88%程度の特性が得られた．①および②の Recall 差も文字/単語各入力 NN の 23%程度に対し 18%程度であった．また，CNN 手法は表 4 に示すとおり，LSTM と比べ学習時間を 1/3 程度にまで削減することが可能である

### 4.3. 実験結果考察

ベースライン手法は，①に対しては学習データと同じ出典の Tweet は全て検出可能であるため，全体の Precision は比較的高く誤検出も少ないが，②に対しては殆ど検出することが不可能であった．このことから，②に対しても 60 %以上もの良好な特性が得られる提案手法全般の有効性が示された．

どの入力系列を用いた場合も，CNN, LSTM 手法は FFNN 手法よりも大幅に高い特性が得られた．このことから，既出・非既出分類のタスクにおいて，時系列情報を学習構造に組み込む事の特性への寄与を確認した．CNN 手法は LSTM 手法と比べ，Precision は高いが，Recall が低いという傾向があり，特に難易度の高いタスクである②に対して文字/単語各入力の NN では 69%程度と，LSTM 手法の 80%以上に比べ，大幅に低い Recall が得られた．この差は，CNN はフィルタサイズ内の時系列のみを考慮し，LSTM は系列全体の時系列を考慮する，構造の違いに起因する．

また，どの提案手法も文字単語結合 NN が最も良好な F 値で，次いで文字，単語という順であった．3.2 節で述べた文字系列の利点が単語系列の利点を上回り，更に文字単語結合 NN では双方の要素が自動的に取捨選択され，より効果的な

学習が可能となったと考えられる. CNN 手法は特に高い改善効果が得られ, LSTM 手法に対して Recall の劣化は 0.6%, ②に対しても 1.2%まで低減した. その結果文字単語結合 NN を用いる場合には, CNN 手法は F 値も LSTM 手法と同程度の特性を得られた.

本システムの現場利用を考慮すると, アノテートの進行による学習データの増加や, 分類の種類が変化する度に学習することも想定され, 学習時間が短いことは利便性が高く, 大きなアドバンテージである. このことから, CNN に文字系列・単語系列を入力し, 出力合成し FFNN で入出力する手法が, 利便性と特性のバランスを得られ, 我々のシステムに最も適している.

**表 2: 提案手法特性評価結果 (%)**

| | | Precision | Recall | F 値 |
|---|---|---|---|---|
| Baseline | | 94.1 | 34.0 | 50.0 |
| 文字入力 NN | FFNN | 92.3 | 78.8 | 85.0 |
| | LSTM | 88.5 | 86.2 | 87.3 |
| | CNN | 94.8 | 78.8 | 86.1 |
| 単語入力 NN | FFNN | 93.5 | 76.2 | 84.0 |
| | LSTM | 90.6 | 82.1 | 86.2 |
| | CNN | 93.7 | 78.4 | 85.4 |
| 文字単語結合 NN (提案手法) | FFNN | 93.3 | 79.6 | 85.9 |
| | LSTM | 91.5 | 84.9 | 88.1 |
| | CNN | **92.8** | **84.3** | **88.4** |

**表 3: テストデータ Recall 比較結果 (%)**

| | | ①出典あり | ②出典なし | 全既出 |
|---|---|---|---|---|
| Baseline | | 76.9 | 3.40 | 34.0 |
| 文字入力 NN | FFNN | 91.7 | 69.6 | 78.8 |
| | LSTM | 95.0 | 79.9 | 86.2 |
| | CNN | 92.1 | 69.4 | 78.8 |
| 単語入力 NN | FFNN | 89.7 | 66.7 | 76.2 |
| | LSTM | 93.3 | 74.2 | 82.1 |
| | CNN | 94.8 | 77.9 | 84.9 |
| 文字単語結合 NN (提案手法) | FFNN | 92.1 | 70.7 | 79.6 |
| | LSTM | 94.8 | 77.9 | 84.9 |
| | CNN | **95.1** | **76.7** | **84.3** |

**表 4: 各 NN 1 epoch 学習時間 (秒)**

| | FFNN | LSTM | CNN |
|---|---|---|---|
| 文字入力 NN | 69.0 | 675 | 86.0 |
| 単語入力 NN | 40.0 | 615 | 284 |
| 文字単語結合 NN (提案手法) | 110 | 1,200 | **440** |

## 5. 関連研究

本稿の関連研究タスクとして, twitter のトピック検出[2], twitter のテキスト二値分類[10], ニュース関連 Tweet の分類[11], ニュース関連記事の分類[12]などが挙げられる. しかし, これらはいずれも抽出関連 Tweet に対して既出・非既出分類するタスクではなく, 第一報抽出と, 第二報以降抽出や意見収集といった本稿の要求に合う分類ではないため, 同タスクの先行研究は存在しない.

また, 本稿の関連研究手法として, 機械学習を用いた NLP において, 単語系列を入力として用い, 補完として文字を用いる NN 構成も提案・評価されてきた[7-8]. しかし, 文字系列は系列全体で一つの意味ベクトル集合であるが, NLP の分類問題において NN で文字系列全体と単語系列全体の出力を結合する手法は著者の知る限り検討されていない.

## 6. おわりに

本稿では, 第一報抽出と, 第二報以降抽出や意見収集といった, 異なる要求に合わせた Tweet 提示システムを構築するため, 既出事項を含む Tweet を抽出するシステム, 及び手法を提案し, 評価した. 特に, 分類処理として文字・単語各入力 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, 88%程度という非常に高い F 値で分類することが可能であるという結果を得た.

しかし, 引用文が文中に入っていないが, ニュースについての意見や, 第二報以降情報が入っている Tweet も多く存在する. そこで今後は, 本稿で提案したシステムの出力を基に, 同じ事柄を話題とする Tweet をまとめるシステム等へ拡張し, 要求する情報の取得をより容易にすることを検討する.

## 文　献


[1] 足立, “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 年.

[2] S. Papadopoulos, et al., “SNOW 2014 Data Challenge: Assessing the Performance of News Topic Detection Methods in Social Media, ” *Proc. SNOW 2014 Data Challenge*, Apr., 2014.

[3] 宮﨑他, “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”, 情処学 IFAT 研報, vol. 127, p.p.1-6, 2017 年 7 月.

[4] 宮﨑他, “ニュース制作のための有用 tweet 掲示システム”, 映情学大, 32B-1, 2017 年 8 月.

[5] D. Bahdanau et al., “Neural Machine Translation by Jointly Learning to Align and Translate, ” *ICLR 2015*, p.p.1-15, Sep., 2014.

[6] Y. Zhang, et al., “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification, ” *IJCNLP 2017*, p.p.253-263, Oct., 2015.

[7] R. K. Srivastava, et al., “Training Very Deep Networks, ” *NIPS2015*, p.p. 2377-2385, Jul., 2015.

[8] X. Ma, et al., “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF, ” *2016 ACL*, p.p. 1064-1074, Mar, 2016

[9] 佐藤他, “単語分かち書き用辞書生成システム NEologd の運用ー文書分類を例にしてー”, 情処学 NL 研報, vol. 15, p.p.1-14, 2016 年 12 月.

[10] S. Rosenthal, et al., “SemEval-2017 Task 4: Sentiment Analysis in Twitter, ” *SemEval-2017*, p.p. 502-518, Aug., 2017.

[11] N. Ghelani et al., “Event Detection on Curated Tweet Streams, ” *SIGIR’17*, p.p. 1325-1328, Aug., 2017.

[12] S. Ribeiro et al., “Unsupervised Event Clustering and Aggregation from Newswire and Web Articles, ” *2017 EMNLP Workshop on Natural Language Processing meets Journalism,* p.p. 62-67, Sep., 2017.