Vol.2020-ICS-199 No.9
2020/3/24

ユーザの過去ツイートを用いた噂の早期検出

小林　鼓[1,a)]　藤田 桂英[2]

概要：インターネットの普及に伴い，twitter 等のソーシャルメディアが発展したことで誰でも簡単に情報発信・収集ができるようになった．これは便利な反面，真実ではない情報や不確実な情報も簡単に広まっていくことに繋がる．自動的に誤った情報を判別する手法としてユーザ間の情報の広まりによる機械学習が有用である．そこで本研究では，情報の広がり方をより正確に認識できるようにするため，ユーザ特徴の表現にユーザのツイート内容を活用する．これにより，学習モデルは効果的に情報の経路を特徴づけることができ，精度向上につながる．また，先行研究に多く見られる噂の構造的側面からのアプローチに比べ，早期に収集できるリツイートユーザの情報は噂の早期検出に貢献する．評価実験には先行研究と同様に噂検出のデータセットである twitter15 を使用し，提案手法で用いるユーザの 2 種類の情報を twitter API を用いて付与した．作成したデータセットにおいて噂検出の精度は向上し，ユーザ特徴を用いた伝播経路による噂検出において過去のツイート内容の使用が有効であることを示した．

1. はじめに

twitter[*1]や facebook[*2]，instagram[*3]といったソーシャルメディアは，インターネットが普及するにつれて人々のコミュニケーションツールとして広く発展してきた．様々な立場の人の情報交換の場として利用され，誰もが情報を発信・受信できるようになった．一方で，このことは悪意のある情報や不確実な情報も広めることにつながる．特に近年では，2016 年のアメリカ大統領選においてトランプ派がソーシャルメディアを活用し人々を扇動するなど，その影響力は社会情勢を揺るがすほどになっている [1]．ソーシャルメディアが社会に与える影響が大きくなるにつれて，人から人に広まっていく情報のコントロールのため，「噂」に関する様々な研究が行われてきた．ここでの噂とは，その真偽を問わずに人々の間で広まっていく話のことである．[2] や [3] では災害時の twitter 上の噂の分析を行い，噂を検出することがユーザの安心に繋がることが示唆されている．

ソーシャルメディアのプラットフォームにおける噂の検出は人手で行われることが多く，新しい情報が増え続けるソーシャルメディアの特性からも，機械による自動化が注目されてきた．自動化の手法は大きく分けて噂の拡散構造に注目して検出する方法と噂に関する最初のツイートに注目して検出する方法の 2 種類であり，現在はどちらの手法においてもニューラルネットワークを用いた研究が主流となっている．拡散構造に基づいた手法では，その噂に関するツイートが十分に集まっていなければならず，検出に時間を要する．噂は人々の目に届く前に検出できることが好ましいため，噂の元ツイートに注目し早期検出した方が有用である．初期段階では元ツイートに関わるユーザ情報が有効とされ，[4] では元ツイートをリツイートしたユーザの特徴ベクトルを時系列に並べ，RNN の入力とした．この研究に用いられたユーザの特徴ベクトルは統計情報だけで作成され，特徴を正確に表せているとはいえない．そこで本研究では，twitter 上の噂の早期検出において，元ツイートをリツイートしたユーザの特徴をより正確に表現することで既存手法の精度向上を図ることを目的とする．twitter 上のユーザ分類問題では，ユーザの twitter における行動を用いることが有用であり，その中でもテキスト情報を用いることが効果的である [5]．リツイートユーザの特徴ごとに元ツイートを分類することと，twitter 上のユーザ分類問題は同義であると仮定し，伝播経路を用いた噂検出におけるテキスト情報の有用性を確かめる．

提案する手法は，「ユーザの統計情報」と「Doc2Vec によりベクトル化したユーザの過去ツイート」の二つの入力から元ツイートを分類する，RNN を二つ組み合わせた 2 入力のモデルである．まず，twitter 上の元ツイートをリツ

[1] 東京農工大学工学部
Faculty of Engineering, Tokyo University of Agriculture and Technology
[2] 東京農工大学大学院工学研究院
Institute of Engineering, Tokyo University of Agriculture and Technology
[a)] kobayashi@katfuji.lab.tuat.ac.jp
[*1] https://twitter.com/
[*2] https://facebook.com/
[*3] https://instagram.com/

イートしたユーザの統計情報と過去ツイートを多変量時系列とみなして，伝播経路をモデル化する．次にこの時系列データを分類するため，長期的な変化まで学習することができる RNN を用いる．統計情報とテキスト情報を同時に一つの RNN に入力し出力を得ることは困難であるため，RNN を二つ用いる．

データセットについて，既存手法に用いられているデータセットは公開されていないが，twitter15[6] というデータセットに twitter API を用いてデータを追加した旨が述べられている．したがって，本研究においても twitter API を用いて統計情報と過去ツイートをそれぞれ取得し新たにデータセットを作成する．

本研究の主な貢献を以下にまとめる．

- 元ツイートの真偽を検出することで未然に誤った情報が流布することを防ぎ，ソーシャルメディアをユーザにとってより安心できるものにする．
- リツイートしたユーザの情報を時系列化した伝播経路を用いることで，噂を可能な限り早期に検出する．
- 既存手法の入力を拡張してユーザの過去ツイートを取り込むことで，より正確な噂検出を行う．

本論文は次のように構成される．まず，噂検出問題とユーザ分類の関連研究を紹介する．次に，本研究で扱う伝播経路を用いた噂の早期検出について，詳細な問題定義とその手法について説明する．次に，提案手法について紹介する. その後，評価実験とデータセットについて述べる. 最後に本研究のまとめと, 今後の課題について論じる.

2. 関連研究

2.1 噂検出

近年，ソーシャルメディアで問題となっている噂 (偽ニュース・誤情報) の検出の自動化に関する研究が注目を集めている．これまでの自動化手法として盛んに行われてきたのが人工的な特徴量を用いた機械学習である．また，噂の流れをネットワーク構造とみなした研究も行われてきた．さらに最近では，ニューラルネットワークの発展に伴って CNN や RNN を用いた研究が注目を集めている [7].

2.1.1 特徴量ベースの機械学習を用いた手法

[8] では，単語の総数やテキストの長さといった統計情報に基づいた特徴抽出を行なった．また，投稿の意味にも注目し感嘆符や疑問符などの感情を表す符号や，単語の極性，URL の有無等についても統計情報により特徴抽出を行なった．テキストの感情分析による噂検出は，[9] でも Linguistic Inquiry and Word Count[*4]という語彙を抽象化してカテゴライズする辞書を用いて行われた．また [9] は感情分析の他に，噂の拡散過程の時間的側面と構造的側面からも特徴抽出を行なった．[10] は投稿の伝播経路を木構造でモデル化して特徴抽出を行うことで，誤った噂の伝播経路は他の噂の伝播経路と異なることを示した．

2.1.2 ニューラルネットワークを用いた手法

前述の [11] や [12] のように，現在では CNN や RNN を使用した手法が注目されている．[13] では，短文の投稿に対処する方法として，CNN と LSTM を用いて投稿の信頼性を出力した．短文の投稿をある時点ごとに一つにまとめて入力としていた従来の手法と比べて，個々の投稿の意味を損うことがないため，噂検出の精度向上に貢献した．[14] はニュース記事の真偽を CNN を用いて判定する際に補助情報として，生成モデルである CVAE を元として作られた “User response generator” による投稿予測を用いた．“User response generator” はソーシャルメディアにおけるニュース記事に対するユーザの投稿を学習したモデルであり，ユーザの真偽判定能力が噂検出の自動化に活用できる可能性を示した．

2.2 ユーザ分類

ソーシャルメディアにおいてユーザを分類することは，ユーザの興味関心に基づいた広告の掲示やおすすめユーザの提案等につながり，またソーシャルボットの検出等にも貢献するため，今日まで様々な研究がなされてきた．[5] は，プロフィール情報，ソーシャルメディア上でのユーザの振る舞い，投稿のテキスト情報，ソーシャルメディア上での他ユーザとのつながりの特徴といった四つの観点から特徴を抽出し，ユーザ分類問題の機械学習による手法を提案した．実験を行なった三つのタスクすべてにおいて，投稿のテキスト情報を用いた特徴が有用であることを示した．[15] はプロフィール情報の統計情報を使用せずにツイート情報等から新たにアノテーションした結果を用いたモデルを提案した．また，[16] はニュースの作成・普及に関わるジャーナリストをソーシャルメディア上で特定するための機械学習モデルを提案し，ジャーナリストの検出におけるユーザ説明文に含まれる情報やフォロワー数，フォロワー比の有用性を示唆した．

3. 噂の早期検出

この章では噂の早期検出の詳細な定義と，提案手法で参考にした [4] で提案された噂早期検出に対するアプローチについて述べる．twitter における噂とは，図 1 のように元ツイートに対してコメントやリツート，いいねといったリアクションがつけられた一連の話の流れのことである．この噂の集合を $\mathcal{A} = \{a_1, a_2, ..., a_{|\mathcal{A}|}\}$ と定義し，a_i に関連するコメントの集合を $\mathcal{M}_i = \{m_1, m_2, ..., m_{|\mathcal{M}|}\}$，ユーザの集合を $\mathcal{U}_i = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ と定義する．a_i はその噂が事実かどうかを表すラベル $L(a_i) = \{0, 1\}^r$ と対応している．例えば本研究で扱うデータセットである twitter15[6] では $r = 2$ であり，それぞれ $L(a_i) = 00$ は a_i が “true” であること，

[*4] https://liwc.wpengine.com/

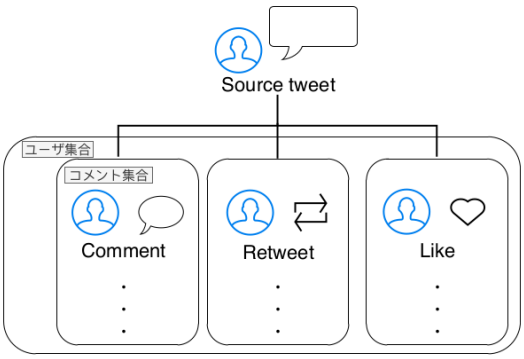


図 1　twitter における噂の構造

$L(a_i) = 01$ は a_i が “false” であること，$L(a_i) = 10$ は a_i が “unverified” であること，$L(a_i) = 11$ は a_i が “non-rumor” であることを表す．噂検出は，メッセージ集合やユーザ集合を元にモデル f を用いて噂のラベルを予測することであり，以下の式で問題を定義できる．

$$L(a_i) = \begin{cases} f(\mathcal{M}_i) \\ f(\mathcal{U}_i) \end{cases} \tag{1}$$

本研究で扱う早期検出とは，噂が発生してから時間が経過しておらず，$\mathcal{M}_i$ や $\mathcal{U}_i$ が少ない状態の a_i のラベルを出力することを指す．ここで，$\mathcal{M}_i$ は構造化できるほど集まっていないが，$\mathcal{U}_i$ は噂の発生初期段階でも比較的集まっており，また個々のユーザの特徴はどの時点でも収集可能であるため，早期検出には後者の方が貢献する．

3.1　ユーザの特徴を用いた伝播経路による早期検出

3.1.1　問題定義

ユーザ集合の有効性から，[4] では以下に説明するように a_i に関わったユーザのベクトル $\boldsymbol{x}_j$ を用いて早期検出に取り組んでいる．まず，ユーザベクトルを用いて，噂 a_i の伝播経路を可変長の多変量時系列 $\mathcal{P}(a_i) = \langle ..., (\boldsymbol{x}_j, t), ... \rangle$ と定義する．ここで，$(\boldsymbol{x}_j, t)$ はユーザ $\boldsymbol{x}_j$ が時間 t において a_i に関するリアクションを起こしたことを意味するタプルである．また，噂に関する最初のツイートが投稿された時刻を $t = 0$ とするため，時系列上では $t > 0$ である．この多変量時系列に基づいて噂検出は式 (2) のように定式化される．

$$L(a_i) = f(\mathcal{P}(a_i)) \tag{2}$$

さらに，早期検出のためには時系列データすべてではなく，一定時間までの伝播経路だけで識別できるモデルである必要があるのでデッドライン T を設けた多変量時系列 $\mathcal{P}(a_i, T) = \langle (\boldsymbol{x}_j, t < T) \rangle$ に基づいて式 (3) と問題を定義する．

$$L(a_i) = f_T(\mathcal{P}(a_i, T)) \tag{3}$$

表 1　先行研究で使用されているユーザのプロフィール情報（[4] を参考に作成）

No.	プロフィール情報	型
1	ユーザ説明文の長さ	Integer
2	ユーザ名の長さ	Integer
3	フォロワーの数	Integer
4	フォローの数	Integer
5	ツイート数	Integer
6	登録からの経過時間	Integer
7	認証の有無	Binary
8	位置情報付与の有無	Binary

3.1.2　ユーザ特徴に統計情報を用いた手法

[4] ではユーザベクトルとして表 1 に示すユーザのプロフィール情報を使用して，多変量時系列を作成する．$\mathcal{P}(a_i)$ が得られたら，それを長さ n の固定長の多変量時系列 $\mathcal{S}(a_i) = \langle \boldsymbol{x}_1, ..., \boldsymbol{x}_n \rangle$ に変換する．変換の際に，$\mathcal{P}(a_i)$ 内のタプルが n より多い場合は最初の n 個を $\mathcal{S}(a_i)$ として扱い，少ない場合は n 個になるまでランダムにオーバーサンプリングする．

作成した固定長の多変量時系列を入力として再帰型ニューラルネットワーク（GRU）と畳み込みニューラルネットワーク（CNN）の二つを用いてラベルの出力を得る．このモデルは，それまでの噂早期検出のどのモデルよりも高い精度を示している．

4.　過去ツイートによるユーザベクトルを用いた噂の早期検出手法

本論文では，ユーザの特徴を用いた伝播経路による噂の早期検出において，ユーザの過去のツイートを用いてユーザの特徴を表現することを提案する．図 2 に概要を示す．twitter API により取得したユーザの過去ツイートを一つの文書とみなし，Doc2Vec を用いてユーザベクトルに変換し，多変量時系列を作成する．また従来手法で使用されている統計情報も同時に入力とするため，新たに GRU を用いた 2 入力 1 出力の学習モデルを用いる．

4.1　ユーザベクトルの作成

4.1.1　ツイッターにおけるユーザ情報の収集

ユーザベクトルを作成するために，twitter API により各ユーザのこれまでのツイートを取得する．twitter API の仕様上ある一定時間内で大量のデータを取得することが困難であるため，本研究で取得するツイートの数は 1 ユーザあたり 100 ツイートとする．また，先行研究において統計情報を用いて作成したデータセットが公開されていないため，統計情報についても取得する必要がある．そのため同様に twitter API を使用して統計情報を取得する．twitter API で得られるプロフィール情報は表 2 に示すとおりで，表 1 で示すもの以外にも取得できる情報がいくつか存在す

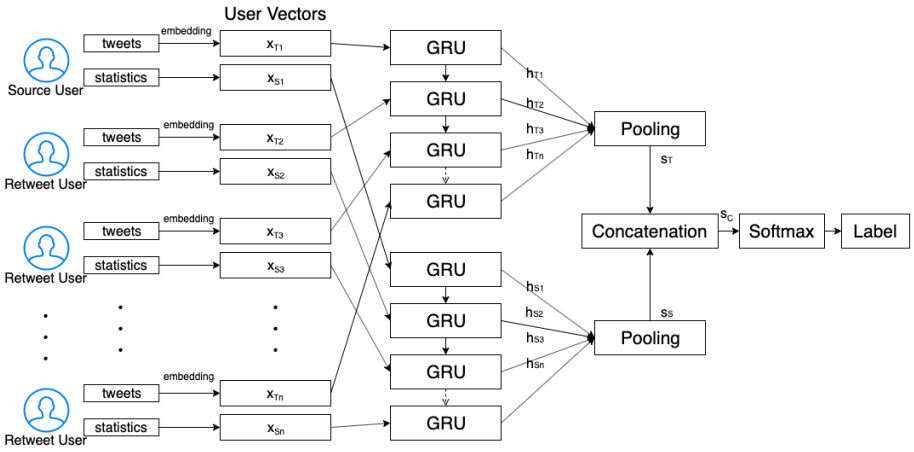


図 2　提案手法の概要

表 2　twitter API により取得できるプロフィール情報一覧 ([17] を参考に作成・一部省略)

返り値	型	説明
name	String	ユーザ名
screen_name	String	@で始まるユーザ名
location	String	ユーザ定義のユーザの場所
url	String	ユーザ定義の url
description	String	ユーザ定義のユーザ説明文
protected	Boolean	非公開設定の有無
verified	Boolean	認証設定の有無
followers_count	Integer	フォロワーの数
friends_count	Integer	フォローの数
listed_count	Integer	ユーザが含まれる公開リストの数
favourites_count	Integer	ユーザがいいねした数
statuses_count	Integer	リツイートも含むツイートの数
created_at	String	アカウント作成時の UTC 日時
default_profile	Boolean	プロフィール画像変更の有無

る．そこで，本研究においては統計情報として従来のものに加えてフォロー／フォロワー比，プロフィール画像の有無，年齢を使用することとする．フォロー／フォロワー比はフォロー数とフォロワー数を用いて計算し，フォロワーが 0 のときは 0 とする．年齢はユーザ定義のユーザ説明文から，[5] で用いられている以下の正規表現により取得し，正規表現とマッチしないとき 0 とする．

```
(I|i)(m|am'|m)([0-9]+)(yo|yearold)
```

4.1.2　Doc2Vec によるベクトル化

4.1.1 項で収集した 100 ツイートを一つにまとめて文書とみなし，Doc2Vec を用いて文書ベクトルとすることでユーザベクトルを作成する．収集したツイートを Doc2Vec の学習に使用し，Doc2Vec の隠れ層の次元は 300 に設定する．各ユーザのツイートを一つの文書としてベクトル化することで，各ユーザのツイートの性質を表した文書ベクトルを作成することができ，統計情報より正確にユーザの特徴を表現することができると考えられる．アカウントを非公開設定にしている等の理由でツイートを取得できないアカウントのユーザベクトルは，同一ラベル内のユーザのベクトルをランダムに選択して使用する．

4.2　GRU を用いた 2 入力 1 出力モデル

ユーザベクトルで構成された伝播経路から噂検出をするための学習モデルとして再起型ニューラルネットワーク (RNN) を応用した Gated Recurrent Unit(GRU)[18] を用いる．GRU ユニットは，伝播経路のユーザベクトル $\boldsymbol{x}_t$ と一つ前の時点における隠れ層 $\boldsymbol{h}_{t-1}$ を入力として，以下の式 4 に従って隠れ層 $\boldsymbol{h}_t$ を出力する．$\boldsymbol{h}_0 = 0$ である．

$$\begin{aligned}
\boldsymbol{z}_t &= \sigma(\boldsymbol{U}_z \boldsymbol{x}_t + \boldsymbol{W}_z \boldsymbol{h}_{t-1}) \\
\boldsymbol{r}_t &= \sigma(\boldsymbol{U}_r \boldsymbol{x}_t + \boldsymbol{W}_r \boldsymbol{h}_{t-1}) \\
\tilde{\boldsymbol{h}}_t &= \tanh(\boldsymbol{U}_h \boldsymbol{x}_t + \boldsymbol{h}_{t-1} \odot \boldsymbol{W}_h \boldsymbol{r}_t) \\
\boldsymbol{h}_t &= (1 - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t
\end{aligned} \tag{4}$$

ここで，$\boldsymbol{U}_z, \boldsymbol{U}_r, \boldsymbol{U}_h \in \mathbb{R}^{m \times d}, \boldsymbol{W}_z, \boldsymbol{W}_r, \boldsymbol{W}_h \in \mathbb{R}^{m \times m}$ は重み行列で，d はユーザベクトル $\boldsymbol{x}_t$ の次元，また m は GRU ユニットの出力の次元である．$\tanh(\cdot)$ はハイパボリックタンジェント関数を表し，$\odot$ は要素積を表す．GRU ユニットによる出力を一つのベクトルにまとめるため，以下の式 5 により平均プーリングを施し，伝播経路の最終的なベクトル表現とする．

$$\boldsymbol{s} = \frac{1}{n} \sum_{t=1}^{n} \boldsymbol{h}_t \tag{5}$$

統計情報と過去ツイートによる二つのベクトルでそれぞれ構成された伝播経路のベクトルを異なる GRU でそれぞれ出力する．式 6 のように，得られるベクトルを連結したものをソフトマックス関数に代入しラベルを出力する．ここで，$\boldsymbol{s}_S$，$\boldsymbol{s}_T$ をそれぞれ統計情報とユーザベクトルで構成された伝播経路のベクトルとする．

表 3　提案手法のモデル構成

ハイパーパラメータ	選択した値	実験範囲 (刻み幅)
GRU hidden size	96	32-128(32)
Adadelta learning rate	0.5	0.1-0.8(0.1)
Adadelta weight decay	0.12	0.1-0.2(0.01)
drop rate	0.5	0.1-0.9(0.1)
batch size	96	16-96(16)

表 4　twitter15 の構成内容

統計項目	総数
true	372
false	370
unverified	374
non-rumor	374
合計	1490

$$\begin{aligned} \boldsymbol{s}_C &= \mathrm{Concatenate}(\boldsymbol{s}_S, \boldsymbol{s}_T) \\ \boldsymbol{l} &= \mathrm{ReLU}(\boldsymbol{s}_C) \\ \boldsymbol{z} &= \mathrm{Softmax}(\boldsymbol{l}) \end{aligned} \tag{6}$$

4.3　モデル構成

提案手法の実装は Python3 で行い，ニューラルネットワークモデル実装のために PyTorch[19] を用いる．モデルは，出力とデータセットのラベルとの損失関数を最小にするように学習する．重みとバイアスの更新には最適化アルゴリズム Adadelta[20] を用いる．過学習を防ぐため，Dropout[21] を各 GRU の隠れ層と GRU の出力の結合層に適用する．学習のエポック数は 200 に設定する．optuna[*5]によりパラメータチューニングを行い，決定したモデルのハイパーパラメータの一覧を表 3 に示す．実験に用いるデータセットのうち 10%をテストデータとして，残りのデータで四分割交差検証を行うこととする．

5.　評価実験

5.1　データセット

5.1.1　使用するデータセット

提案モデルを評価するために，twitter 上の実データにより構成される噂検出のためのデータセットである twitter15[6] を用いる．このデータセットはソースツイートをリツイートしたユーザにより伝播経路が構成されている．ラベルは “true”，“false”，“unverified”，“non-rumor” の四つであり，“true”・“false” は噂が「真」か「偽」かを表し，“unverified” は「噂の真偽が不明」であることを表す．また，“non-rumor” はソースツイートが「噂ではない」情報に関するツイートであることを示す．データセットの統計量を表 4 に示す．公開されているデータセットはユーザ情報が含まれていないため，twitter API でプロフィール情報と過去ツイートを収集する．

[*5] https://preferred.jp/ja/projects/optuna/

表 5　追加情報の取得結果

統計項目	総数
データセット内のユーザ総数	368,113
プロフィール情報を取得できたユーザ数	330,619
多変量時系列のユーザ総数	102,923
ツイート情報を取得できたユーザ数	81,596

5.1.2　追加要素の収集結果

ユーザ情報を取得した結果について，表 5 に示すようになった．まずソースツイートをリツイートしたユーザについて，twitter API によりプロフィール情報を取得したが，twitter15 内のユーザのうち約 4 万人分のデータが取得できなかった．得られたプロフィール情報で $n = 100$ として多変量時系列を作成した結果のユーザ総数が表 5 に示す多変量時系列のユーザ総数となった．作成した多変量時系列のユーザベクトルに付与する形で twitter API によりツイート情報を取得したが，非公開等の理由で約 1 万ユーザのうち 20%ほどのユーザのツイートを取得することができなかった．

5.1.3　過去ツイートに対する前処理

ユーザのツイートを使用するにあたり，学習に影響を与える単語や文字を取り除いた．具体的には，どんなツイートでも登場しうる一般的な表現や単語として意味を成さないものの削除を行い，Doc2Vec の単語埋め込みにおけるユーザの特徴の学習効率を高めた．まずツイートをすべて小文字に変換し，手動で引用リツイートする際に用いる “rt” やリプライツイートの先頭に表示されるユーザ名である “@screen_name”，記事やウェブサイトの URL を削除した．次に，無駄な改行や空白，記号等を削除し，絵文字も削除した．また，一文字以下の単語は意味をなさないと判断し削除した．最後に，ストップワードと呼ばれる，一般にすべての文章に頻出する単語群を消去した．出現頻度が多い単語は単語分散表現の精度を落とすと言われており，これを除去することでより効率よく文書ベクトルの学習が進むようにした．ストップワードの除去には自然言語処理用のライブラリである NLTK[*6]を用いた．

5.2　実験設定

5.2.1　比較モデル

本実験では提案手法の，噂検出におけるユーザ特徴にツイート情報を用いることの効果を検証するため，従来手法であるプロフィール情報を用いたモデルとの比較を行なった．新たに取得した統計情報を追加した場合の効果も検証するため，以下に示すユーザ特徴とモデルの組み合わせで実験を行なった．

既存手法：　ユーザ特徴はプロフィール情報だけを使用，学習モデルに GRU+CNN を使用．

[*6] https://www.nltk.org/

表 6　twitter15 における噂検出結果 (それぞれのラベルの列は F1 値を表す．)

手法	精度	true	false	unverified	non-rumor
既存手法	0.518	0.393	0.402	**0.553**	0.763
既存手法+apf	0.516	0.409	0.375	0.494	**0.777**
提案手法	**0.529**	**0.530**	0.281	0.476	0.712
提案手法+apf	0.526	0.505	0.356	0.527	0.667
提案手法 −pf	0.521	0.420	**0.452**	0.475	0.724

既存手法+apf：　ユーザ特徴はプロフィール情報と新たに追加したプロフィール情報 (added profile features) を使用，学習モデルに GRU+CNN を使用．

提案手法：　ユーザ特徴はプロフィール情報とツイート情報を使用，学習モデルに GRU を用いた 2 入力 1 出力のモデルを使用．

提案手法+apf：　ユーザ特徴はプロフィール情報と新たに追加したプロフィール情報とツイート情報を使用，学習モデルに GRU を用いた 2 入力 1 出力のモデルを使用．

提案手法 −pf：　ユーザ特徴にプロフィール情報 (profile features) を使用せずツイート情報だけを使用，学習モデルに GRU+CNN を使用．

5.2.2　伝播経路の固定長 n の設定

評価実験の入力に用いる，多変量時系列の固定長 n について，既存研究の結果に基づいて設定した．[4] では twitter15 において，$n = 40$ となると学習モデルの精度は収束し，n をそれ以上大きくしてもほとんど変化がないことが示唆された．そこで本実験では，n に起因する精度の向上が十分に収束していると判断できる $n = 50$ とした．

5.3　実験結果

実験に用いたすべてのモデルのテストデータに対する精度と F1 値は表 6 のようになり，提案手法であるツイート情報とプロフィール情報を用いたモデルが最も高い精度を示した．既存手法と既存手法+apf，提案手法と提案手法+apf をそれぞれ比べるとほぼ変わらない精度となり，プロフィール情報をこれ以上増やしても精度向上は見込めないことが示された．既存手法と提案手法を比べると，提案手法が既存手法を上回り，ツイート情報を含めたユーザベクトルを用いることで精度が向上することが示された．以上のことから，プロフィール情報だけでユーザの特徴を表現し伝播経路を構成するには限界があり，より正確な伝播経路を作成するにはツイート情報を使用した方が良いことが示唆された．

提案手法 −pf は既存手法と同等の精度になり，提案手法と比べると精度が低くなった．このことから，プロフィール情報はユーザを特徴付けるには足りないが，不必要ではないことが示された．ユーザの特徴をより正確に表すには，プロフィール情報とツイート情報の二つを用いるべきであり，それらで構成される伝播経路が最も噂検出に貢献することが確認できた．

各ラベルの F1 値を見ると，提案手法は既存手法と比べて “true” のスコアが高くなっており，ツイート情報を用いると “true” である噂の伝播経路をより特徴づけしやすくなることが示された．逆に，“false” のスコアが既存手法と比べて低くなってしまった．図 3 に示す混同行列においても同様の予測正誤の様子が確認され，既存手法と比べて提案手法は “true” を正しく予測できたが，“false” を “true” と予測する傾向が強かった．しかし，‘true” と “false” の予測正誤の様子を見ると，既存手法では “true” を “false” と誤って予測してしまった数が提案手法より多くなっており，伝播経路の分類精度の観点では，提案手法が勝っているといえる．

5.4　議論

5.4.1　データセットに対する精度の低さ

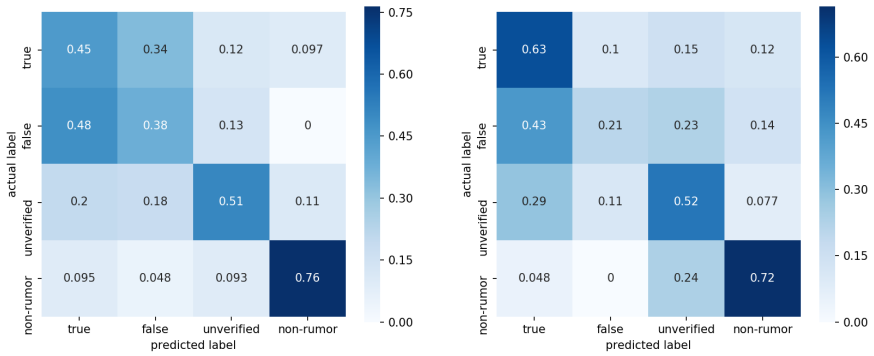
本研究に用いたデータセット twitter15 を用いた先行研究における精度は，表 7 に示すとおりで，本実験で実装したモデルは既存手法 [4] の精度を再現できていない．最も大きな要因としてあげられるのが，経年によるデータセットのユーザデータの損失である．表 5 で示すように，既存手法のプロフィール情報取得時点で約 4 万件のユーザ情報が損失しており，データセットを完全に再現することができなかった．不完全なデータセットではモデルは正しい伝播経路を学習できずに，精度に影響を与えたと考えられる．また，既存研究ではユーザ情報を取得するスクリプトを公開していないため，twitter API による情報取得の実装が異なっていた可能性も考えられる．

提案手法であるツイート情報の取得でもユーザデータの損失が問題となっている．約 2 万件のツイート情報が取得できていない状態でユーザベクトルを作成し伝播経路を構築したため，本来の伝播経路と異なった伝播経路を構築してしまった可能性が否定できない．そのため，より実データに即したデータセットを用いることで提案手法の精度向上が見込める．

6.　おわりに

6.1　まとめ

本論文では，ユーザの特徴を用いた伝播経路による噂の



[1] 既存手法　　[2] 提案手法

図 3　提案手法と既存手法の混同行列

表 7　先行研究における精度 ([4] を参考に作成)

手法	精度
GRU	0.646
RFC	0.565
PTK	0.750
PPC_RNN	0.811
PPC_CNN	0.803
PPC_RNN+CNN	0.842

早期検出において，ユーザの特徴として新たにユーザ分類問題で有用とされている過去ツイートを用いることを提案した．また，同時にユーザベクトルに統計情報を入れた場合と入れなかった場合の精度の比較も行い，統計情報がユーザの特徴を表すのにどの程度貢献しているかを確認した．

噂検出に用いられるデータセット twitter15 に，twitter API により必要なユーザの情報を付与し実験を行った結果から，噂の伝播経路作成の際のユーザ特徴として過去ツイートを用いることは有用であることと，ユーザを特徴づける情報としてはプロフィール情報とツイート情報の両方を用いた方が良いことが示された．

6.2　今後の課題

提案手法は，ユーザ分類の分野で有用な過去ツイートを用いており，実験結果からもその有効性が示されている．これは，ユーザの特徴を用いた伝播経路の分類におけるユーザの特徴の定義とユーザ分類問題が同義であることを裏付ける結果である．したがって，ユーザ分類の手法をより詳しく調査して噂の伝播経路構築に役立てることが今後期待される．

本研究はソーシャルメディアとして代表的な twitter に焦点を当てたが，その他にも facebook や instagram 等，ユーザ情報を利用できるメディアは多数存在する．本研究により過去の投稿を利用したユーザ特徴を用いた伝播経路の構築が情報の信憑性を確認するのに有効な手段であると確かめられたので，他のメディアへの応用が期待される．

参考文献

[1] Craig, S.: This Analysis Shows How Viral Fake Election News Stories Outperformed Real News On Facebook, (online), available from ⟨`https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook`⟩ (accessed 2020-01-30).

[2] Valecha, R., Oh, O. and Rao, R.: An exploration of collaboration over time in collective crisis response during the Haiti 2010 earthquake (2013).

[3] Mendoza, M., Poblete, B. and Castillo, C.: Twitter under crisis: Can we trust what we RT?, *Proceedings of the first workshop on social media analytics*, ACM, pp. 71–79 (2010).

[4] Liu, Y. and Wu, Y.-F. B.: Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks, *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).

[5] Pennacchiotti, M. and Popescu, A.-M.: A machine learning approach to twitter user classification, *Fifth International AAAI Conference on Weblogs and Social Media* (2011).

[6] Liu, X., Nourbakhsh, A., Li, Q., Fang, R. and Shah, S.: Real-time Rumor Debunking on Twitter, *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 1867–1870 (2015).

[7] Cao, J., Guo, J., Li, X., Jin, Z., Guo, H. and Li, J.: Automatic rumor detection on microblogs: A survey, *arXiv preprint arXiv:1807.03505* (2018).

[8] Castillo, C., Mendoza, M. and Poblete, B.: Information credibility on twitter, *Proceedings of the 20th international conference on World wide web*, ACM, pp. 675–684 (2011).

[9] Kwon, S., Cha, M., Jung, K., Chen, W. and Wang, Y.: Prominent features of rumor propagation in online social media, *2013 IEEE 13th International Conference on Data Mining*, IEEE, pp. 1103–1108 (2013).

[10] Wu, K., Yang, S. and Zhu, K. Q.: False rumors detection on sina weibo by propagation structures, *2015 IEEE 31st international conference on data engineering*, IEEE, pp. 651–662 (2015).

[11] Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K.-F. and Cha, M.: Detecting rumors from microblogs with recurrent neural networks., *Ijcai*, pp. 3818–3824 (2016).

[12] Chen, T., Li, X., Yin, H. and Zhang, J.: Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, pp. 40–52 (2018).
[13] Nguyen, T. N., Li, C. and Niederée, C.: On early-stage debunking rumors on twitter: Leveraging the wisdom of weak learners, *International Conference on Social Informatics*, Springer, pp. 141–158 (2017).
[14] Qian, F., Gong, C., Sharma, K. and Liu, Y.: Neural User Response Generator: Fake News Detection with Collective User Intelligence., *IJCAI*, pp. 3834–3840 (2018).
[15] Volkova, S., Bachrach, Y., Armstrong, M. and Sharma, V.: Inferring latent user properties from texts published in social media, *Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015).
[16] Zeng, L., Dailey, D., Mohamed, O., Starbird, K. and Spiro, E. S.: Detecting Journalism in the Age of Social Media: Three Experiments in Classifying Journalists on Twitter, *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 13, No. 01, pp. 548–559 (2019).
[17] : User object — Twitter Developers, （オンライン）, 入手先 ⟨`https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object`⟩ （参照 2020-01-13）.
[18] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation, *arXiv preprint arXiv:1406.1078* (2014).
[19] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. et al.: PyTorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems*, pp. 8024–8035 (2019).
[20] Zeiler, M. D.: ADADELTA: an adaptive learning rate method, *arXiv preprint arXiv:1212.5701* (2012).
[21] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting, *The journal of machine learning research*, Vol. 15, No. 1, pp. 1929–1958 (2014).