

Graph Convolutional Networkを用いた ソーシャルメディア上の偽情報の早期発見

須藤 広平[†] 張 建偉[†]

[†] 岩手大学大学院総合科学研究科 〒020-8551 岩手県盛岡市上田 4-3-5

E-mail: [†]{g0320080,zhang}@iwate-u.ac.jp

あらまし 近年のソーシャルメディアの普及と共に、ソーシャルメディア上の偽情報の問題が深刻化している。ソーシャルメディアが普及し、あらゆる世代の多くの人々が利用する現代において、偽情報が社会に与える影響は大きい。例えば、近年ではコロナウイルスに関する偽情報が多く錯綜し、社会を混乱させた。ソーシャルメディア上の情報は膨大で、拡散が非常に早いことが特徴であり、人間による監視や検出は困難である。そこで、本研究では Graph Convolutional Network (GCN) という深層学習モデルとユーザー特徴・テキスト特徴を用い、ソーシャルメディア特有のネットワーク構造とその変化を考慮した深層学習モデルの学習を行う偽情報の早期検出手法を提案する。

キーワード ソーシャルメディア、時系列データ、グラフデータ、機械学習、偽情報、早期検出、GCN

1 はじめに

近年、ソーシャルメディアは人々の生活の中に急速に普及してきている。図1は2020年に総務省によって行われた「令和2年通信利用動向調査」の結果を参考に作成した、日本における年齢層別のソーシャルネットワーキングサービス (SNS) 利用状況のグラフである [1]。これを見ると、全体で73.8%の人がソーシャルメディアの1つであるSNSを利用しており、さらに、あらゆる世代の人がSNSを利用していることも分かる。また、調査結果の中では、62.1%の人がSNSの利用目的について「知りたいことについて情報を探すため」と回答したと報告されている [1]。このように多くの人々が利用し、情報を検索するソーシャルメディアは利便性が高く、あらゆるトピックの情報源としても有用である。

しかしながら、同時に偽情報の拡散が深刻さを増している。例えば、近年では新型コロナウイルスに関する様々な情報の錯綜が記憶に新しい。総務省ではインターネット上に蔓延する新型コロナウイルスに関する偽情報の実態についても調査を行い、結果をまとめている。調査結果の中では、ソーシャルメディアの1つであるTwitter¹で新型コロナウイルスに関する偽情報があたかも真実または真偽不明の情報として書かれているのを目にしたと答えた人の割合が57.0%と高い割合を示していた [2]。つまり、ソーシャルメディアは人々の情報源として利用されることが多いにもかかわらず、知らず知らずのうちに偽情報を取得する可能性も十分にあるということが言える。このような状況を考慮すると、ソーシャルメディア上の偽情報への対応は急務であるが、ソーシャルメディア上の膨大で拡散が非常に早い情報を人間が逐一確認するというのは現実的ではない。また、偽情報を早期に発見し、迅速な対応を行うということも偽情報の拡散を防ぐ上で重要な点である。

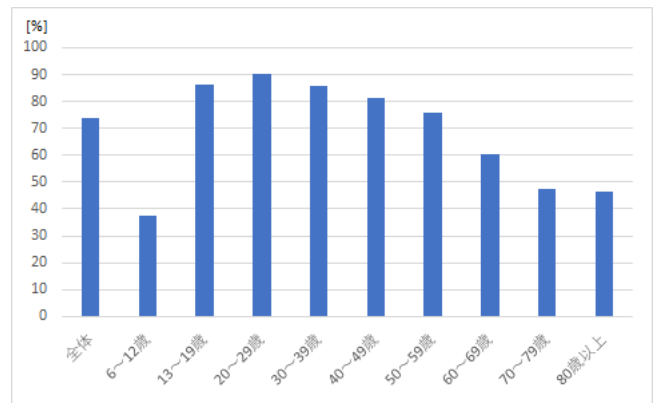


図1 2020年のSNS個人利用状況 ([1]を参考に作成)

以上のような背景から、ソーシャルメディア上の偽情報を機械的かつ早期に検出する技術が求められている。そこで本研究では Graph Convolutional Network (GCN) [3] を用いてソーシャルメディア特有のネットワーク構造とその変化の両方を考慮したグラフのベクトル表現獲得を行うと共に、ユーザー特徴量だけでなく、偽情報検出において重要な特徴量の1つであると考えられるテキスト特徴も入力として使用する偽情報の早期検出手法を提案する。また、本研究では、ソーシャルメディアの1つであり、有名なSNSであるTwitterより収集されたデータセットを使用し、本提案手法について学習率ごとやエポック数ごと、最大シーケンス長ごとによる性能比較とプーリング層の違いによる性能比較の検証を行った。

本論文の構成は以下の通りである。第2章では既存の偽情報検出手法について紹介する。第3章では我々の提案手法について詳細を述べ、第4章では評価実験の結果を報告する。第4章の結果をもとに第5章では考察を述べ、第6章で本論文の結論をまとめる。

¹ : <https://twitter.com>

2 関連研究

Bondielli ら [4] はフェイクニュースやうわさなどの用語の定義からデータセットの収集方法やその問題点、検出方法のアプローチや使用された特徴量などこの分野について幅広く調査を行い、まとめている。Meel ら [5] は Bondielli らと同様に用語の定義や偽情報検出のアプローチなどについて触れつつ、こちらは世界的に人気のある SNS プラットフォームや情報汚染のサイクルに注目した言及を行っている。Lillie ら [6] は主張のスタンス分類がフェイクニュース検出にもつながるとして、フェイクニュース検出の方法だけでなく、スタンス分類のアプローチについても触れている。また、データ収集法や特徴量抽出、データアノテーションの方法も述べている。

Castillo ら [7] は Twitter 上において収集できるテキストやユーザ情報などに基づいた非常に多くの特徴量を使用し、Twitter 上においてニュース性のあるトピックの検出と、検出したトピックに関して J48 決定木を用いた真偽判定を試みる論文を発表している。また、偽情報の検出において決定木を用いた有用な特徴量の分析や特徴量同士の相関性の検証など偽情報検出のアプローチだけでなく、特徴量に関する幅広い考察も行い、非常に多くの知見を発表した。

また、Zhang ら [8] は SVM などの伝統的な機械学習手法と MLP を使用し、データの時系列変化とイベントの発生規模を考慮した効率的な偽情報の早期検出手法を提案した。

Castillo ら [7] と Zhang ら [8] の研究は特徴量の作成段階ではソーシャルメディア特有の情報の伝播経路やネットワーク構造を考慮しているが、モデルの学習段階では考慮していないという課題点がある。また、伝統的な機械学習を中心に用いているが、本研究では深層学習を用いるという点でも異なる。

近年ではモデルの学習段階においても伝播経路やネットワーク構造を考慮した研究が行われており、モデルの学習段階においてもソーシャルメディア特有の情報伝播という側面が注目されている。その 1 つとして、Liu ら [9] は情報の伝播経路を考慮した偽情報検出を試みている。この先行研究ではソーシャルメディア上のユーザ情報を入力とし、RNN (GRU) と CNN を用いた特徴量の抽出を行うことで、グローバルなデータの変化とローカルなデータの変化の両方を取り入れた特徴量を利用した偽情報の早期検出手法を提案し、その有効性を示した。

また、Bian ら [10] はネットワーク構造を考慮した偽情報検出を提案している。この先行研究では Graph Convolutional Network (GCN) をベースに情報のトップダウンの流れとボトムアップの流れを考慮した特徴量の抽出手法と、抽出の際にグラフのルートとなるノードの特徴量を各層の出力の後に追加することで偽情報検出において重要と考えられるルートノードの特徴量の影響を強める手法の 2 つを合わせた手法を提案した。この提案手法を用い、偽情報検出においてトップダウンの伝播とボトムアップの集約を考慮することやルートノードの影響を強めることの有効性を確認している。

Liu ら [9] の研究では深層学習と伝播構造を用いて偽情報検

出を試みているが、ネットワーク構造そのものを考慮することはできていない。また、入力にはユーザの特徴のみを用いており、偽情報検出において重要だと考えられるユーザの考えや感情、表現が含まれやすいテキスト特徴量を用いていない。一方、Bian ら [10] の研究ではネットワーク構造は考慮できているものの、ネットワーク構造の変化については考慮していない。

本研究では、ソーシャルメディア特有のネットワーク構造とその変化の両方を考慮した特徴量抽出を行うと共に、ユーザ特徴だけでなく、偽情報検出において重要な特徴の 1 つであると考えられるテキスト特徴も入力として使用するという点でこれらの先行研究とは異なっている。

3 提案手法

提案手法の全体像を図 2 に示す。本提案手法は、ユーザ特徴側とテキスト特徴側の 2 つに分けて特徴量抽出を行う。この特徴量抽出には Kipf ら [3] が考案した GCN を用いる。その後、抽出した特徴量を結合し、全結合層へ入力するという仕組みになっている。

3.1 問題設定

今回偽情報検出の対象とするソーシャルメディアは日本でも有名な SNS の 1 つである Twitter とする。また、偽情報の検出に関しては情報の出所となったツイート² (ソースツイート) の真偽について予測を行うものとする。つまり、本研究の場合、ソースツイートに対する予測は情報拡散によって構築されたネットワーク構造や情報拡散に関与したユーザ (ソースツイートの投稿者も含む) の情報、情報拡散の際に投稿されたテキスト (ソースツイート、リプライ³、リツイート⁴) の情報を考慮する。また、本実験では最大シーケンス長を設定することで使用できるデータ数に上限を設ける。このシーケンス長とは、伝播回数+1 (リプライ・リツイート+ソースツイート) に等しいものである。

3.2 Graph Convolutional Network (GCN)

本節では、本研究で使用する Graph Convolutional Network (GCN) の概要について説明する。本研究で使用する GCN は Kipf ら [3] によって 2017 年に発表されたグラフ構造を考慮した畳み込みを行うことができる深層学習モデルである。この深層学習モデルはグラフ構造を入力として受け取り、対象ノードとエッジを持つノード (隣接するノード) との特徴量の畳み込み演算を、1 層中でグラフのすべてのノードを対象に行う (図 3)。隠れ層でこのような演算を行った結果、最終的に各ノードがグラフ構造を考慮したベクトル表現を獲得する。また、層を重ねて使用することで対象ノードから離れたノードに関しても間接的に特徴量を取り込むことができる。さらに、各ノードが獲得したベクトル表現を読み出すことでグラフ構造を考慮したグラフのベクトル表現を獲得することもできる。

2: ユーザが投稿するテキスト

3: ツイートに対する返信のこと

4: ツイートを拡散すること

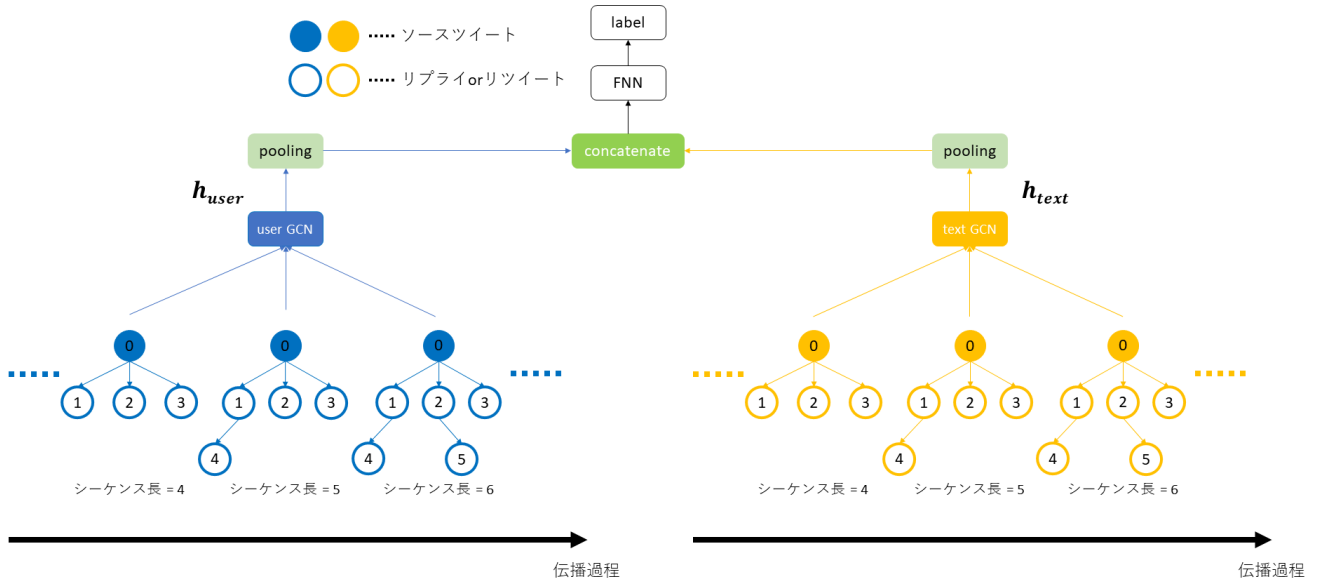


図2 提案手法の全体像

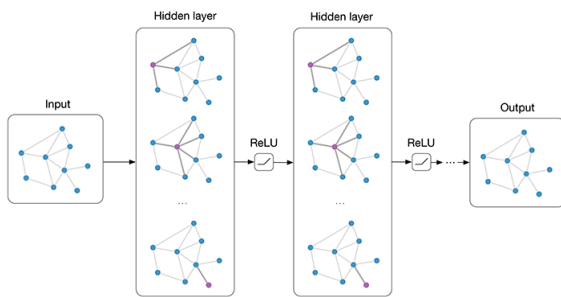


図3 GCNの概要[3,11] (出典:[11])

本研究ではこのGCNを使用し、ノードごとにベクトル表現を獲得した後、各ノードが獲得したベクトル表現を平均化することによってグラフ単位のベクトル表現を獲得する。これにより、ソーシャルメディア特有のネットワーク構造を考慮したグラフ単位のベクトル表現を獲得することができるため、情報の拡散状況をより反映させた特徴量抽出を行うことができると考えている。

3.3 シーケンス長ごとのグラフ構築

GCNへ入力する際に、シーケンス長ごとのグラフ構築を行う。入力の際に最大シーケンス長を定め、最大シーケンス長までのデータをもとに、シーケンス長ごとにグラフ構築を行う。このようにして構築されたシーケンス長ごとのグラフをユーザー特徴側のGCNとテキスト特徴側のGCNに入力する。この際、ユーザー特徴側GCNに入力されるグラフのノードはユーザー特徴量を持ち、テキスト特徴側GCNに入力されるグラフのノードはテキスト特徴量を持つ。この際、留意しなければならないのは最大シーケンス長を長くすればするほど使用できるデータも増えるが、予測や学習に必要なデータも増えるということである。1つのソースツイートに対して、シーケンス長ごとのグラフを構築し、作成された複数のグラフをひとまとまりとしてGCNに入力する。

本研究ではこのように情報の拡散状況に応じたネットワーク構造の変化をこのシーケンス長ごとのグラフ構築で考慮することにより、現実世界の情報拡散の流れを反映させた学習ができるのではないかと考えている。また、シーケンス長に応じて、情報拡散の初期段階から中期、後期段階まで考慮することができる上に、最大シーケンス長を短く設定することで少ないデータでも学習や予測が可能のため、偽情報の早期検出にも貢献できるのではないかと考えている。

3.4 GCNによるグラフのベクトル表現獲得と集約

3.2節で述べた通り、グラフを構築した後、GCNによるグラフベクトル表現の獲得を行う。GCNはノードの特徴量を隣接ノードとのエッジの有無に応じて畳み込むという性質を持っている。この畳み込み処理を行った後に、各ノードが獲得したベクトル表現を読み出し、平均化することでグラフのベクトル表現を獲得する。これをすべてのシーケンス長のグラフで行うことにより、それぞれのシーケンス長に対応したグラフの構造を反映したグラフのベクトル表現を獲得できる。このようにして獲得したグラフのベクトル表現に対し、さらに平均プーリングによる集約を行うことで、それぞれのシーケンス長ごとのグラフから獲得したグラフのベクトル表現を平均化したものを最終的に得ることができる。

本研究では、このようにGCNを利用し、ネットワーク構造を考慮したグラフのベクトル表現獲得とすべてのシーケンス長のグラフを対象としたグラフのベクトル表現の集約を行うことで、グラフ構造の変化を総合的に考慮した特徴を抽出することができると考えている。

3.5 テキスト情報の利用

本研究では、ユーザー情報だけでなく、ユーザーが投稿したテキストにまつわる情報も利用する。とあるトピックやニュースに対してユーザーが反応し、投稿したテキストにはユーザーの考え

表1 データセットの詳細

名称	値
true ラベル数	372
false ラベル数	368
平均シーケンス長	342.7
シーケンス長の中央値	203
最大シーケンス長	2,979
最小シーケンス長	88

や感情、表現が含まれていることが多い。時間経過とともに、ソースツイートに対する否定的な意見や肯定的な意見、懐疑的な意見など様々な意見が多くなると考えられるため、本提案手法にテキスト情報も用いることで時間経過による人々の考えや感情の変化も反映させることができるのではないかと考えている。また、ソースツイートのテキスト内容も偽情報の検出には重要な要素であると言える。以上のような理由から、本研究ではテキスト情報についても重要視しており、偽情報検出の精度向上のために必要な情報なのではないかと考えている。

4 評価実験

4.1 データセット

今回の実験では Ma ら [12] が作成した Twitter15 と呼ばれるデータセットを使用する。Twitter15 データセットは Twitter を対象に収集したデータセットである。データセットはツイート ID とユーザ ID、投稿の遅延時間、伝播経路で構成されており、真偽予測対象であるソースツイートとそれに対するリプライやリツイートなどの情報がセットで含まれている。この Twitter15 データセットをもとに Twitter API⁵を用いて、ユーザ情報とツイートテキストの収集を行った。今回の実験で使用するデータセットの詳細に関しては表 1 に示す。なお、データセットのラベルとしては“true”、“false”、“unverified”、“non-rumor”の4種類が存在するが、今回はそのうち“true”または“false”のどちらかのラベルをもつデータのみを採用している。また、一部データに関してはグラフ生成が困難である等の理由で使用していないため、若干元のデータ数よりも少なくなっている。

4.2 特徴量の作成

4.2.1 特徴量の種類

GCN に入力する特徴量の作成に関しては、Castillo ら [7] や Zhang ら [8]、Liu ら [9]、Ma ら [13] の先行研究を参考に本研究の偽情報検出において有効であると考えられる特徴量を検討した。その結果、表 2 に示す、ユーザベース特徴量 7 種、テキストベース特徴量 7 種の計 14 種の特徴量を作成した。表 2 の特徴量のうち、感情値とポジティブ・ネガティブな単語の割合に関する特徴量の作成方法については次項以降で詳細を述べる。

4.2.2 感情値

テキストベース特徴量の 1 つであるツイートの感情値に関しては VADER [14] を使用して算出を行った。VADER は人間の

表2 特 徴 量

カテゴリ	特徴量
ユーザベース (7 次元)	ユーザ説明欄の記述の長さ
	アカウント名の長さ
	フォロワー数
	フォロー数
	総ツイート数
	アカウント作成からの期間
	位置情報許可の有無
テキストベース (7 次元)	ツイートの感情値
	URL の有無
	? の個数
	! の個数
	# (ハッシュタグ) の個数
	ツイート内におけるポジティブな単語の割合
	ツイート内におけるネガティブな単語の割合

評価者によって評価された 7,000 語以上の感情スコアを持った単語辞書と文法・構文的な観点から導き出された 5 つのルールを用いて感情の強さの修正を行うことにより感情値を算出する感情分析ツールである。顔文字やネットスラングなどにも対応していることから本研究の実験対象となっている Twitter などのようなソーシャルメディア上のテキストに対する感情分析に対して特に適している。

本研究では VADER が分析した値のうち、文章の各単語ごとの感情スコアを考慮し、正規化した値である Compound の値を感情値として使用する。

4.2.3 ポジティブ・ネガティブな単語の割合

本研究では各ツイートに用いられている単語に対して、それぞれポジティブ・ネガティブなどの感情による判別を行う。それぞれの単語の感情判別については AFFIN-en-165 [15] と呼ばれる感情辞書を使用する。AFFIN-en-165 はインターネットスラングなどを含む 3,300 語以上の単語に対して感情強度を定義した感情辞書である。これを用いて各単語の感情判別を行い、各ツイートごとにポジティブな単語が使用されている割合、ネガティブな単語が使用されている割合を算出し、特徴量として使用する。

4.3 学習モデルの作成とパラメータの設定

本研究では、実装する際のプログラミング言語として Python⁶ を使用した。また、ユーザ特徴側とテキスト特徴側でそれぞれ GCN が必要となる。今回は深層学習ライブラリである PyTorch⁷ と Deep Graph Library⁸ [16] を使用して実装を行った。

また、使用したパラメータ、活性化関数等については表 3 と表 4 にて示す。パラメータや活性化関数等の設定に関しては筆者の経験則に基づいて設定を行った。なお、学習率とエポック数、最大シーケンス長、プーリング層については実験によって異なるため、実験の際に都度説明を加えることとする。

6 : <https://www.python.org/>

7 : <https://pytorch.org/>

8 : <https://www.dgl.ai/>

5 : <https://developer.twitter.com/en/docs/twitter-api>

表3 パラメータ

パラメータ	設定値
バッチサイズ	16
GCN 層数	2
GCN 出力次元	20
全結合層数	1
全結合層出力次元	2

表4 活性化関数等

隠れ層活性化関数	ReLU
出力層活性化関数	Sigmoid
最適化アルゴリズム	Adam
損失関数	Binary Cross Entropy

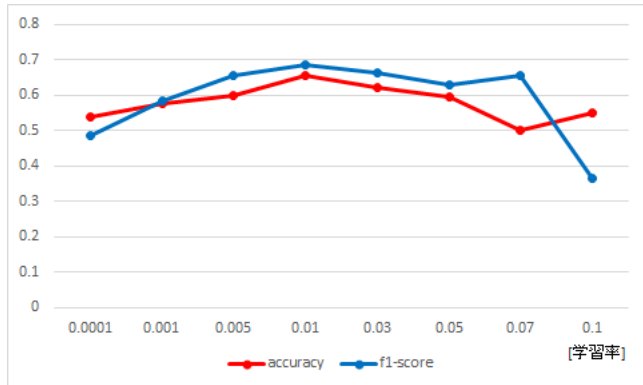


図4 学習率ごとの性能比較結果

4.4 実験

4.4.1 学習率ごとの性能比較

まず、学習率ごとの性能比較実験を行う。この実験により、学習率による性能への影響を検証する。学習率は0.0001, 0.001, 0.005, 0.01, 0.03, 0.05, 0.07, 0.1の8種類で比較を行う。パラメータについてはエポック数を60, 最大シーケンス長を50とし、プーリング層についてはmean poolingを採用した。検証方法はhold-out法とし、トレーニングデータ数が75%, テストデータ数が25%の割合となるように分割し、実験を行う。なお、検証データについては今回はデータセットがあまり大きくない関係で使用しないこととした。

実験結果について、図4に示す。図は学習率ごとの精度 (accuracy) とF値 (f1-score) の変化を折れ線グラフで表したものとなっている。グラフを見てみると、学習率が0.01に近づくにつれて精度、F値共に上昇し、学習率0.01を超えた付近からやや性能が下がる傾向にあるという結果となった。特に、学習率が0.1の際にはF値が大きく下がる結果となってしまった。また、今回の実験における最大精度、F値は、共に学習率が0.01の際の0.65と0.69であった。

4.4.2 エポック数ごとの性能比較

本項ではエポック数による性能比較実験を行う。この実験により、エポック数による性能への影響を検証する。エポック数は10~100までの10刻みで実験を行い、性能を比較する。パラメータについては学習率を0.01, 最大シーケンス長を50と

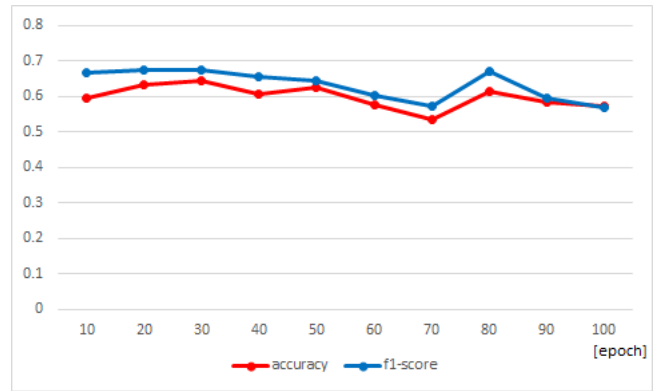


図5 エポック数ごとの性能比較結果

し、プーリング層についてはmean poolingを採用した。検証方法はhold-out法とし、前項と同じくトレーニングデータ数が75%, テストデータ数が25%となるように分割し、検証データについては使用しないという条件で実験を行った。

実験結果について、図5に示す。図はエポック数ごとの精度とF値の変化を折れ線グラフで表したものとなっている。グラフを見てみると、エポック数が増えるほど精度、F値共にやや下降気味になっている。精度とF値の両方に関して、エポック数が30の時点で最大に達し、その後は全体的に下降気味である。また、今回の実験において最大精度と最大F値を達成したのは共にエポック数が30の時であり、それぞれ0.64と0.68であった。

4.4.3 最大シーケンス長ごとの性能比較

本項では、最大シーケンス長による性能比較実験を行う。この実験により、本提案手法の偽情報の早期検出性や少数データによる学習・予測性能を検証する。最大シーケンス長は10~100の10刻みで実験を行い、性能を比較する。パラメータについては学習率を0.01, エポック数を60とし、プーリング層についてはmean poolingを採用した。検証方法はhold-out法を採用し、前項までと同様にトレーニングデータ数が75%, テストデータ数が25%, 検証データはなしという条件で実験を行った。

実験結果を図6に示す。精度に注目すると、最大シーケンス長が20から30にかけての部分で急上昇し、その後40, 50はほぼ平坦となっている。最大シーケンス長が50を超えたところで急激に精度が下がり、その後は少々持ち直してはいるものの最大シーケンス長が30~50付近ほどの精度にはならないという結果になった。次にF値に注目する。今回の実験ではF値は概ね0.62~0.66の付近に収まっているが、最大シーケンス長が60の時のみ極端に値が小さくなるという結果になった。なお、今回の実験における最大精度は最大シーケンス長が50のときの0.64であり、最大F値は最大シーケンス長が70のときの0.67であった。

4.4.4 プーリング層の違いによる性能比較

本項では、プーリング層の違いによる性能比較を行う。本研究では基本的にプーリング層としてmean poolingを用いているが、本項ではそのプーリング層をmax poolingに変更した場合どのような結果が得られるのかを検証する。パラメータについ

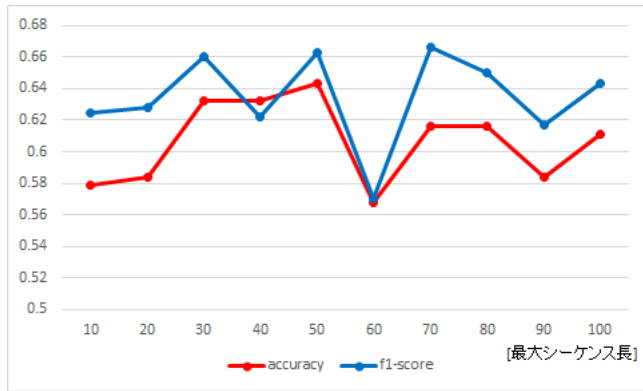


図6 最大シーケンス長ごとの性能比較結果

表5 プーリング層の違いによる性能比較

プーリング層	mean pooling	max pooling
精度 (accuracy)	0.6000	0.6054
適合率 (precision)	0.5862	0.6235
再現率 (recall)	0.7234	0.5638
F 値 (f1-score)	0.6476	0.5922

ては学習率を 0.01, エポック数を 60, 最大シーケンス長を 50 とした。また, 検証方法については前項までと同様に hold-out 法を採用し, トレーニングデータ数が 75%, テストデータ数が 25%, 検証データはなしという条件で実験を行った。

実験結果を表 5 に示す。双方を比較すると, 精度と適合率に関しては max pooling が勝る結果となった。一方, 再現率と F 値に関しては mean pooling が勝る結果となった。この結果について, どちらの方が性能が良いということは明確には言い難いが, 精度に関する結果は 2 つのプーリング層の間でごくわずかな差であるのに対し, F 値は 2 つのプーリング層の間である程度の差が生じている。このことを考慮すると, どちらかと言えば mean pooling の方が性能が良かったと言えるのではないだろうか。

4.4.5 テキスト特徴量の有無による性能比較

最後に, テキスト特徴量の有無による性能比較を行う。この実験により, テキスト特徴量の有無が性能にどのような影響を与えるかを検証する。パラメータについては学習率を 0.01, エポック数を 60, 最大シーケンス長を 50 とし, プーリング層については mean pooling を採用した。検証方法については, 層化 4 分割交差検証を採用し, 合計 4 回のテスト結果の平均を最終的な結果として使用する。また, ユーザ特徴量のみの場合図 2 のユーザ特徴側の構造のみを使用し, プーリング層を通した後, そのまま全結合層 (FNN) へ入力する。

実験結果を表 6 に示す。結果を比較すると, 全体的にわずかながらユーザ特徴量とテキスト特徴量の両方を使用した場合の方が性能が上回る結果となった。

5 考察

5.1 本提案手法の偽情報検出への有効性

まず, 本提案手法の偽情報検出への有効性について検討する。

表6 テキスト特徴量の有無による性能比較

特徴量	user+text	user only
精度 (accuracy)	0.55	0.54
適合率 (precision)	0.55	0.54
再現率 (recall)	0.53	0.50
F 値 (f1-score)	0.54	0.52

表7 ツイートデータの詳細

取得を試みたツイート数 (重複なし)	26,352
取得できたツイート数	18,651
取得できなかったツイート数	7,701
取得できなかったツイートの割合	0.29

図 4～図 6 を見ると, 精度や F 値が 0.6 を超える場面が見られるが, 精度, F 値共に 0.7 を超えることはできなかった。また, 層化 4 分割交差検証を用いた表 6 の結果を見ると, 精度が 0.55, F 値は 0.54 と決して高い有効性を示せたとは言えない結果となってしまった。

このような結果となってしまった原因は大きく 2 つ考えられる。

1 つ目はデータセットの規模である。今回の実験では使用したデータセットの中で “true” または “false” のラベルが付けられているデータのみを使用したということもあり, ソースツイートのデータ数が 720 データと決して大きいとは言えないものであった。そのため, より大規模なデータセットを用いて, 学習データを増やすことができれば, さらなる性能を発揮できるのではないかと考えている。

2 つ目はデータ収集の段階における収集不可能なデータの存在と欠損値処理の問題が考えられる。Twitter15 データセットが Ma ら [12] の研究で作成・使用され, 論文として発表されたのは 2017 年である。そのため, 本研究で Twitter15 データセットをもとに TwitterAPI を用いたデータ収集を行う段階で, データの収集対象となっているユーザが既に Twitter を退会していたり, 収集対象のツイートそのものが削除されていたりすることなどが原因で収集することができなかったと考えられるデータが存在した。特に, ツイートの削除に関してはユーザの退会よりも発生する可能性が高いと考えられ, 実際に本研究で使用したデータセットを調査したところ表 7 のような結果が得られた。この表を参考にすると, 取得できなかったツイートの割合が 0.29 と決して低い割合ではない。さらに, 今回の実験では欠損値をすべて 0 に置き換えることで補完を行っていたことも高い精度や F 値を得ることができなかった原因の 1 つではないかと推測される。

5.2 本提案手法の偽情報の早期検出性

次に, 本提案手法の偽情報の早期検出性について検討する。図 6 を参考に考察すると, 今回の実験では最大シーケンス長が 30 の時点で精度 0.6 を超え, F 値も 0.65 を超えることができた。これについては表 1 よりシーケンス長の中央値が 203, 平均シーケンス長が 342.7 であることを考慮すると, 少ないデータ数でもある程度の学習と予測が可能であることが示せたと

考えられる。しかし、図6について、筆者の予想では最大シーケンス長が大きくなるほど使用できるデータが増えるため、性能は向上すると考えていたが、結果としては図6の通り、最大シーケンス長が60の際に精度とF値が大幅に低下するなど不安定な結果となってしまった。このような結果となってしまった理由としては5.1節と同じく、取得不能データの存在と欠損値処理の影響があるのではないかと考えている。表7の通り、取得できなかったツイートの割合は0.29と決して小さくはない。そのため、最大シーケンス長が大きくなった際に、取得できなかったツイート、つまり欠損値となっているデータが多数追加入力されることも考えられる。これが今回の実験において、モデルの性能が不安定になってしまった原因なのではないだろうかと推測している。

5.3 特徴量にテキスト特徴量を使用することの有効性

最後に、本提案手法において、特徴量にテキスト特徴量を使用することの有効性について検討する。表6を見ると、精度に関しては1%ほどの違いしかないが、再現率については3%、F値については2%の違いがある。この結果から、多少なりともテキスト特徴量が本提案手法のモデルの性能向上に影響を与えているのではないかと考えられる。今回の実験ではテキスト特徴量を使用することによって、わずかな性能向上という結果しか得ることができなかったが、表7を参考にすると、本来ならばあと約3割ほどテキスト特徴量を加えることが可能であった。そのため、今回取得できなかったツイートを取得することができていた場合、今回の結果よりもさらに性能を向上させることができていたのではないかと考えられる。

6 まとめと今後の展望

本研究ではGCNを用いてソーシャルメディア特有のネットワーク構造とその変化の両方を考慮したグラフのベクトル表現獲得を行うと共に、ユーザ特徴量とテキスト特徴量の両方を入力として使用する偽情報の早期検出手法を提案した。また、本論文を通して、本提案手法の有効性について様々な観点からの検証を行い、最大シーケンス長に関する実験によって、本提案手法の早期検出性を検証した。加えて、テキスト特徴量の有無による性能比較を行うことによって、本提案手法においてテキスト特徴量を加えることの有効性の検証を行った。

結果としては、本提案手法は高い性能を示せたとは言いがたい結果となってしまった。しかしながら、最大シーケンス長が短い場合の学習や予測であっても、ある程度の性能が発揮できることを示せたと考えられる。また、テキスト特徴量を使用することによる有効性についてもわずかではあるが、性能が向上するという結果を得ることができた。

今後の展望としては、データの欠損値に対して平均値や中央値などの統計値を用い、適切な欠損値処理を行うことでさらなる性能の向上を見込めるのではないかと考えている。また、今回よりもさらに大規模なデータセットを用いることによる性能の検証やプーリング層における特徴量集約の方法に工夫を施す

ことも考えられる。最後に、本研究で使用したGCN以外にもグラフ構造を考慮した深層学習モデルは様々な発表されており、これまでもGraph Attention Network (GAT) と呼ばれるアテンション機構を搭載したモデル [17] や GraphSAGE と呼ばれるフレームワーク [18] が提案されている。そのようなモデルについてもさらに調査を行い、より高い性能と早期検出性能を得るために今後も偽情報の早期発見に取り組んでいきたい。

謝 辞

本研究はJSPS 科研費 19K12230 の助成を受けたものである。

文 献

- [1] 総務省, “令和 2 年通信利用動向調査の結果”, https://www.soumu.go.jp/johotsusintokei/statistics/data/210618_1.pdf, 2022-02-10 参照.
- [2] 総務省, “新型コロナウイルス感染症に関する情報流通調査”, https://www.soumu.go.jp/main_content/000693280.pdf, 2022-02-10 参照.
- [3] Thomas N. Kipf, Max Welling, “Semi-Supervised Classification with Graph Convolutional Networks”, ICLR 2017, 2017.
- [4] Alessandro Bondielli, Francesco Marcelloni, “A Survey on Fake News and Rumour Detection Techniques”, Information Sciences, vol.497, pp.38-55, 2019.
- [5] Priyanka Meel, Dinesh Kumar Vishwakarma, “Fake News, Rumor, Information Pollution in Social Media and Web: A Contemporary Survey of State-Of-The-Arts, Challenges and Opportunities”, Expert Systems with Applications, vol.153, 2020.
- [6] Anders Edelbo Lillie, Emil Refsgaard Middelboe, “Fake News Detection using Stance Classification: A Survey”, CoRR abs/1907.00181, 2019.
- [7] Carlos Castillo, Marcelo Mendoza, Barbara Poblete, “Information Credibility on Twitter”, WWW 2011, pp.675-684, 2011.
- [8] Jianwei Zhang, Jinto Yamanaka, Lin Li, “Early Automatic Detection of False Information in Twitter Event Considering Occurrence Scale and Time Series”, iiWAS 2020, pp.282-289, 2020.
- [9] Yang Liu, Yi-fang Brook Wu, “Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks”, AAAI 2018, pp.354-361, 2018.
- [10] Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, Junzhou Huang, “Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks”, AAAI 2020, pp.549-556, 2020.
- [11] Thomas N. Kipf, “Graph Convolutional Networks”, <https://tkipf.github.io/graph-convolutional-networks/>, 2022-02-10 参照.
- [12] Jing Ma, Wei Gao, Kam-Fai Wong, “Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning”, ACL 2017, pp.708-717, 2017.
- [13] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, Kam-Fai Wong, “Detect Rumors Using Time Series of Social Context Information on Microblogging Websites”, CIKM 2015, pp.1751-1754, 2015.
- [14] Clayton J. Hutto, Eric Gilbert, “VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text”, ICWSM 2014, 2014.
- [15] Finn Årup Nielsen, “A New ANEW: Evaluation of A Word List for Sentiment Analysis in Microblogs”, #MSM 2011, pp.93-98, 2011.
- [16] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, Zheng Zhang, “Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks”, arXiv preprint arXiv:1909.01315, 2019.
- [17] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, Yoshua Bengio, “Graph Attention Networks”,

ICLR 2018, 2018.

- [18] William L. Hamilton, Zhitao Ying, Jure Leskovec, “Inductive Representation Learning on Large Graphs”, NIPS 2017, pp1024-1034, 2017.