Twitter 固有の特徴を考慮した噂に関する投稿のスタンス分類

高田　大輔†　　杉山　一成†

† 京都大学情報学研究科　〒 606–8501 京都市左京区吉田本町
E-mail: †takata.daisuke.72r@kyoto-u.jp, ††kaz.sugiyama@i.kyoto-u.ac.jp

あらまし　ソーシャルメディアは，膨大な情報にアクセスすることができる便利なツールである一方，そこで拡散された誤情報が大きな社会的問題を引き起こすことがある．誤情報の拡散に対処するには，情報の真偽が定まっていない「噂」の段階で，その真偽を分類することが重要である．噂の信憑性は，噂に寄せられた投稿の噂に対するスタンスと相関関係にあることが知られているため，そのスタンスは「噂の真偽判定」にとって非常に重要な役割を果たす．本研究は，この「噂に関する投稿のスタンス分類」に焦点を当て，既存研究が抱えるテキストデータを十分に扱えていないという問題点を改善する手法と，関連ツイートと Twitter 固有の機能から得られる情報を活用する手法を提案し，それらの有効性を示した．有効性が確認された全ての提案手法を用いたモデルは，本研究で使用するデータセットを用いたコンペティションにおいて最も高いスコアを達成した．

キーワード　Twitter，ソーシャルメディア，誤情報，カプセルネットワーク

1 はじめに

今日，ソーシャルメディアは，多くの人々が情報収集のために用いる一般的なツールとなっている．複数あるプラットフォームの中の 1 つである Twitter は，各ユーザが「ツイート」と呼ばれる短いテキストを自由に投稿することができ，情報を収集・拡散するためのプラットフォームとして人気を博している．ソーシャルメディアを使用することで，膨大な情報にアクセスすることができるというメリットがある一方，誤った情報が拡散されていき，それが大きな社会的問題を引き起こすことがある．近年では，新型コロナウイルスに関する誤情報によって，各地で混乱が生じたことが記憶に新しい．社会の混乱につながるような誤情報の伝播を防ぐために，噂の真偽を分類する数多くの研究が行われている [1].

噂の分類の研究は大きく分けて 4 つのステップから成り立つ．1 つ目のステップである「(1) 噂の検出」は，一連の投稿からどの投稿が噂の投稿，もしくは噂に関して議論している投稿かを判断するものである．2 つ目のステップである「(2) 噂の追跡」は，ある投稿が噂に関する投稿であると特定された (もともと噂であるとされていた，もしくは前述の噂の検出のステップで特定された) 後に，議論している噂の種類に従って分類し，クラスタを形成するものである．3 つ目のステップである「(3) 噂に関する投稿のスタンス分類」は，特定の噂に関連した各投稿が，その噂の真実性に対してどのようなスタンスを持っているのかを決定するものである．スタンスには，“Support”・“Query”・“Deny”・“Comment” の 4 種類が存在する．スレッドでの議論における噂の真実性の予測に着目した研究 [2] [3] により，噂の信憑性は，その議論に参加している投稿の噂に対するスタンスと相関関係にあることが示されたので，このステップは，後続の「噂の真偽判定」のステップを容易にする働きを持つ．これは言い換えれば，「群衆の知恵」を利用することで噂の真偽を判定するという事である．4 つ目のステップである「(4) 噂の真偽判定」は，噂と判定されたものが真であるか，偽であるか，またはその真偽がまだ明らかになっていない未検証のものであるかを判定するものである．

誤情報の拡散に対処するには，上述の 4 つのステップを早期に達成することが求められる．本研究では，「(3) 噂に関する投稿のスタンス分類」のタスクに焦点を当て，Twitter 上の噂に関する投稿のスタンス分類の既存研究が抱えている問題点を指摘し，それを改善するような手法，ならびに，精度の向上が期待できるような追加の手法を提案する．各手法の有効性を確認した後，それらを利用したモデルを構築し，本研究が扱うデータセットが用いられたコンペティションの結果と比較する．

2 関連研究

2.1 カプセルネットワーク

カプセルネットワーク [4] は，従来の Convolutional Neural Netowrk (CNN) の欠点を克服し，CNN を拡張したモデルである．従来の CNN には，プーリングの段階で位置不変性を獲得すると同時に，空間的な構造の情報を喪失してしまうという問題点があった．この問題点から，例えば画像処理タスクでは，人間の目で見ると明らかに位置関係がおかしい画像が正しいものと認識されるケースや，1 つのオブジェクトに対してあらゆる角度から撮った画像が必要となるケースがあった．この問題を解決するべく，ニューラルネットワークにおいてニューロンがスカラーを扱うのに対し，カプセルネットワークは，空間情報をベクトル化した「カプセル」という新しい概念を扱う．表 1 は，ニューロンとカプセルの違いをまとめたものである．

それぞれのカプセルは，画像や文章中のオブジェクトを表すもので，オブジェクトがどのように見えるかを表す特徴を保有している．結合したカプセルは，それぞれのカプセルが表すものの関係を表しており，これによって画像や文章内の構造や関

表 1　カプセルネットワーク内のカプセル (左) とニューラルネットワーク内のニューロン (右) の違い.

	カプセル	ニューロン
入力，出力の形式	ベクトル	スカラー
Affine 変換	$\hat{\boldsymbol{u}}_{\boldsymbol{j}\boldsymbol{\mid}\boldsymbol{i}} = \boldsymbol{W}_{ij}\boldsymbol{u}_i$	-
重みづけと合計	$\boldsymbol{s}_j = \sum_i c_{ij}\hat{\boldsymbol{u}}_{\boldsymbol{j}\boldsymbol{\mid}\boldsymbol{i}}$	$a_j = \sum_i w_i x_i + b$
非線形な活性化関数	$\boldsymbol{v}_j = \frac{\|\|s_j\|\|^2}{1+\|\|s_j\|\|^2}\frac{s_j}{\|\|s_j\|\|}$	$h_j = f(a_j)$

係性を表す事ができる．そのため，カプセルネットワークは他のニューラルネットワークよりも複雑で構造化された画像や文章を，精度良く分類することができる．

2.2　自然言語処理モデル

近年の言語モデルには，深層学習アルゴリズムを活用し，高い精度を実現するものが多い．“Bidirectional Encoder Representations from Transformers” (BERT) [5] は，Transformer という強力なニューラルネットワークを使用した自然言語処理モデルである．Attention という手法を用いて離れた位置にある情報も適切に取り入れることができ，文脈を考慮して単語の分散表現を生成する．学習には，大量のデータを用いて汎用的な言語のパターンを学習する「事前学習」の段階と，比較的少数のデータを用いて特定のタスクに特化するよう学習する「ファインチューニング」の段階が存在する．英語版の BERT では，事前学習に 33 億語が用いられた．“Robustly Optimized BERT Pretraining Approach” (RoBERTa) [6] は，BERT をベースとして，追加で 145GB のデータを事前学習に用い，また，学習方法をより機能的なものに改良したモデルである．

BERT や RoBERTa のように一般の文章を中心に用いて事前学習を行ったモデルとは違い，事前学習にツイートを用いたモデルがいくつか存在する．TweetEval [7] は，最近の Twitter 関連のタスクには，標準となるタスク・モデル・ベースラインが存在しないために，モデルの性能を測りづらいという問題を解決するべく，ベースとなるタスクやモデルを作成した研究である．TweetEval で作成された言語モデルは，RoBERTa をベースとして，追加で 2018 年 5 月から 2019 年 8 月までの 5840 万ツイートを事前学習したものである．本研究で Twitter-RoBERTa という名前で実験に用いたモデルがこれにあたる．BERTweet [8] は，大規模な量のツイートをゼロから学習したモデルである．ツイートは規範的な言語基準を持つメディアとは違い，非公式な文体やノイズの多い情報が含まれるため，ツイートを用いた学習を行うタスクでは，通常の文章で事前学習したモデルよりも，あらかじめツイートを事前学習したモデルの方が適しているという発想のもと生まれた．BERTweet は BERT をベースとしたアーキテクチャを持っており，事前学習の手順は，RoBERTa のそれと同様である．BERTweet はゼロから 2012 年 1 月から 2020 年 3 月までの 8.5 億もの英語のツイートを用いて学習を行なった．

3　提 案 手 法

本研究では，Twitter 上の噂に関する投稿のスタンス分類の既存研究が抱える問題を解決する手法，ならびに，精度向上に繋がると考えられる 2 つの手法を提案する．本章では，それら 3 つの手法の詳細に加え，有効性が確認された各手法を活用したモデルの構築についても詳述する．

3.1　ツイートのテキストデータが持つ問題点への対処

ツイートを扱う研究において，テキストデータは必ず使われるものであるが，これを通常のテキストと同様に扱うには，2 つの問題点が存在する．1 つ目は，ツイートは必ずしも規則正しい文法に則って書かれた文体ではないということである．新聞や雑誌，書籍などの，規範的な文体に沿って書かれた文章とは違い，ツイートというのは，ユーザーが個人で自由に書いた文章である．そのため，スペルミス，非公式な文法，ソーシャルメディア上の文章に特有の言い回し，絵文字などの，ノイズとなるような情報を多く含む．それ以外にも Twitter では「メンション機能」や「ハッシュタグ機能」などの，ツイートに特殊な文字を組み込むことによって使うことができるようになる機能が存在し，これらもノイズとなり得る．2 つ目は，ツイートの文字数の少なさである．プラットフォームの特性上，ツイートには文字数制限が存在し，既存のデータセット上では，日本語では 140 字，英語では 280 字となっている．文字数の少なさは，テキストそのものから得られる情報量が少ないという問題に直結する．

3.1.1　Twitter に特化した自然言語処理モデルの利用

本研究では，Twitter に特化した言語モデルを文章のトークン化・単語の分散表現の獲得に用いることを提案する．BERT のような，規範的な文章で事前学習された言語モデルではなく，ツイートを事前学習に用いた言語モデルを使用することで，ノイズを含み，かつテキスト長が短いツイートの表現を学習しやすいと考えた．そこで，2.2 節で取り上げた 4 つの言語モデル（BERT，RoBERTa，Twitter-RoBERTa，BERTweet）を用いて学習を行い，モデル間で精度を比較する．BERT，RoBERTa は規範的な文章を中心に事前学習を行ったモデルであり，Twitter-RoBERTa は RoBERTa をベースとして追加でツイートを学習したモデル，BERTweet はゼロからツイートを学習したモデルである．各モデルでツイート全体の分散表現を獲得し，得られた分散表現のうちの [cls] トークンに対応する分散表現を用いて分類を行い，精度を比較する．モデルの全体像を，図 1 に示す．

3.1.2　ツイート全体の分散表現を利用

言語モデルの出力として得られたツイート全体の分散表現のうちの [cls] トークンに対応する分散表現が，出力を集約したものであるために，これを用いて文章分類を行う場面が多く存在する．本研究では，[cls] トークンに対応する分散表現を用いた分類の結果に加えて，ツイート全体の分散表現を用いた分類の結果を合わせるモデルを提案する．ツイート全体の分散表現も

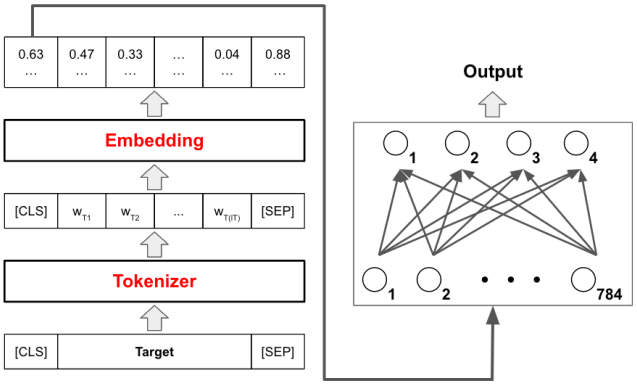

図 1　[cls] トークンに対応する分散表現を用いて分類を行うモデルの全体像．図中の赤文字で示した Tokenizer，Embedding に用いる言語モデルを入れ替え，各モデルの効果を比較検証する．図中，Target はターゲットツイートを表す．なお，図中の数値は例として示したものであり，これは以降のモデル図においても同様である．

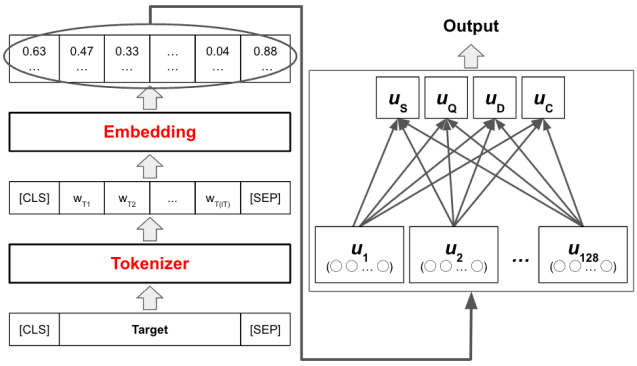

図 2　ツイート全体の分散表現を用いてカプセルネットワークで分類を行うモデルの全体像．図 1 同様，図中の赤文字で示した Tokenizer，Embedding に用いる言語モデルを入れ替えることで，各モデルの効果を比較検証する．

また分類に利用できると考えたのは，出力を集約したものである [cls] トークンに対応する分散表現を分類に用いることができるということは，集約前の全体の分散表現もうまく扱うことができれば，分類に用いることができる，さらには集約された分散表現では失われている特徴がある程度存在していると考えられるため，集約前の分散表現を用いることは精度の向上に繋がりうると考えたためである．ツイート全体の分散表現を用いた分類には，CNN の諸問題を解決することを目的として作られた分類器である，カプセルネットワークを利用する．ツイート全体の分散表現を各トークンごとにカプセルに入れて分類することによって，言語モデルが生み出した密な特徴表現を活かした結果が得られると考えた．[cls] トークンに対応する分散表現を用いた分類の結果に加えて，ツイート全体の分散表現を用いた分類の結果を合わせたモデルを構築する前に，2.2 節で取り上げた 4 つの言語モデルとカプセルネットワークを組み合わせたモデルを構築し，最適な言語モデルを選択する．そのためのモデルの全体像を，図 2 に示す．

3.2　ターゲットツイートに関連するツイートの利用

スタンス分類のターゲットとなるツイート (以降，Target Tweet，“TT” と略記する) に深く関連したツイートとして，ソースツイート (以降，Source Tweet，“ST” と略記する)・TT の親ツイート・TT の子ツイートが存在する．ST は TT のトピックのようなものとして捉えることができ，TT の親ツイートや子ツイートは，TT が直接スタンスを示す対象，もしくは示される対象である．文脈を考慮して単語の分散表現を得るという BERT (とそれから派生する言語モデル) の特性から，TT に関連した文脈を持つそれらのツイートを利用することが，精度向上に繋がると考えた．ただし，TT の子ツイートは複数存在し得るものであり，その数は様々であるため，本研究では ST と TT の親ツイートを学習に利用する．なお，ST の親ツイートは存在しないため，その場合の親ツイートには ST を繰り返す形で利用した．

追加の 2 つの関連ツイートは，文の区切りを示すトークン [SEP] と未使用のトークン [unused0] を用い，言語モデルに入力として加えた．[unused0] トークンを用いた理由は，BERT が [SEP] トークンを用いての 3 文以上の入力に対応していないためである．「関連ツイートを利用しなかったモデル」と「関連ツイートを利用したモデル」の 2 つのパターンで比較実験を行った．なお，言語モデルの出力のうちの [cls] トークンに対応する分散表現を用いた分類と，ツイート全体の分散表現を用いたカプセルネットワークによる分類の，どちらのパターンにおいても，この比較実験を行った．「関連ツイートを利用しなかったモデル」は，図 1 と図 2 に示したものと同様である．「関連ツイートを利用したモデル」の全体図を図 3 に示す．

3.3　Twitter 固有の機能から得られる情報の利用

Twitter 固有の機能として代表的なものに，リツイート機能 (以降，RT と略記する)，いいね機能，フォロー機能が存在する．「RT」や「いいね」は，ツイートが持つ意見に対してなんらかの感情を持った際に行う行為であるため，RT 数やいいね数が多いツイートというのは，多数の人にとってそのスタンスが比較的明白なものであると考えられる．そこで，ツイートのスタンスとの関連が考えられるこれら 2 つの機能から得られる数値を分類に利用することを考えた．また，フォロイー数・フォロワー数（以降，FF 数と略記する）に関しても，「よく意見が支持される人はフォロワー数が多い」「否定や疑問ばかりする人はフォロワー数が少なく，フォロイーの方が多い傾向にある」などの，ツイートのスタンスとの関連が考えられるので，FF 数も分類に利用することを考えた．具体的には，言語モデルの出力のうちの [cls] トークンに対応する分散表現に続く形で Twitter 固有の機能から得られた数値を特徴として加えることで，精度の向上を図る．なお，Twitter 固有の機能から得られた数値を特徴として加える際には，標準化を行った．この手法の有効性を確認するべく，「追加の特徴を利用しなかったモデル」，「FF 数を追加の特徴として利用したモデル」，「RT 数・いいね数を追加の特徴として利用したモデル」，「FF 数・RT 数・いいね数を追加の特徴として利用したモデル」の 4 つ

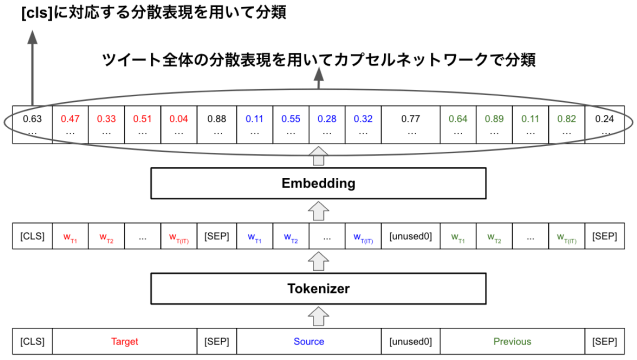

図 3　TT の関連ツイートを分類に利用する際の概略．図中，Target はターゲットツイート，Source はソースツイート，Previous はターゲットツイートの親ツイートを表す．トークナイザーによるトークン化をしたのちの単語 w の添字は出現順を表し，lT・lS・lP はそれぞれ Target・Source・Previous をトークン化した後のトークン長を表している．分散表現を用いた分類の様子は図 1，2 と同様なため，略記した．

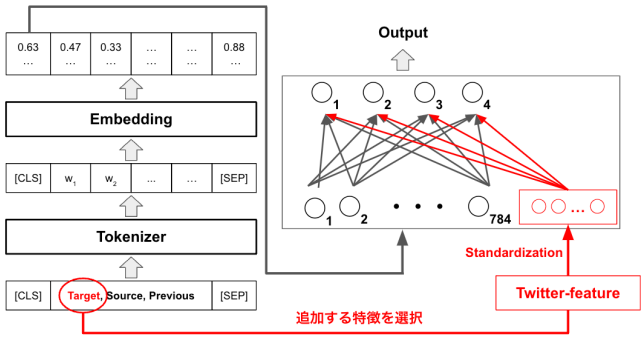

図 4　Twitter 固有の機能から得られる情報を利用して分類をするモデルの全体像．入力として 3.2 節で提案した関連ツイートの追加がされているが，これは本モデルを構築した際に，既に関連ツイートの利用の有効性が確認されていたためである．

のパターンで比較実験を行った．この実験を行ったモデルの全体図を図 4 に示す．

3.4　アンサンブル学習 (スタッキング) の利用

本研究における実験には，言語モデルから得られた出力のうち，[cls] トークンに対応する分散表現を用いるものと，ツイート全体の分散表現を用いるものの 2 パターンが存在し，それぞれにおいて異なる分類器を用いる，そこで，最終的にそれらのモデルの強みを共有するべく，アンサンブル学習を行う．アンサンブル学習には，いくつかの手法が存在する．本研究では，複数のモデルを用いて予測値を算出し，その予測値を新たな特徴量として再び学習を行うスタッキングの手法を選択する．スタッキングは単体モデルよりも精度が向上することが多いが，結果の解釈・分析が難しい．したがって，前節までに述べた各提案手法は，単独のモデルを用いて効果検証を行っている．また，スタッキングは性質の異なる分類器を複数織り交ぜることで，いろいろな長所を取り入れることができるため，分類器として，Transformer やカプセルネットワークなどのニューラルネットの性質を持つ分類器と性質の異なる，ロジスティック回帰・サポートベクターマシン・ランダムフォレスト・k 近傍法・勾配ブースティングを加えた．最終的なモデルを図 5 に示す．

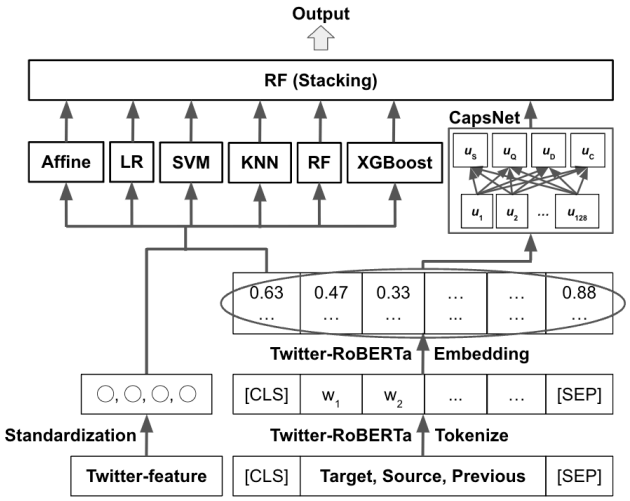

図 5　スタッキングの手法でアンサンブル学習を行う最終的なモデルの全体像．言語モデルとして Twitter-RoBERTa が用いられ，また，3.2 節と 3.3 節で提案した手法がどちらも適用されているが，これは本モデルを作成した時点で各手法の有効性が確認されていたためである．なお，図中の LR はロジスティック回帰，SVM はサポートベクターマシン，KNN は k 近傍法，RF はランダムフォレストを表している．図中の上部に示したように，スタッキングの最終段階の学習にはランダムフォレストを用いた．

4　実験準備

4.1　データセット

本研究では，SemEval - 2017 Task 7 : RumourEval の subtaskA [9] (以降，RumourEval と略記する) のデータセットを用いて実験を行う．表 2 に，データセットの内容を示す．このデータセットは，データセットが作られた当時に話題となっていた Twitter 上での 9 つのトピックに関する議論から収集されたツイートから構成される．噂のツイートから始まり，それに対してのリプライツイートが続く一連の会話スレッドを，各トピックが複数個含んでおり，合計 297 個の会話スレッドが存在する．

表 2　SemEval2017 のデータセット．

	Support	Query	Deny	Comment	Total
train	841	330	333	2,734	4,238
dev	69	28	11	173	281
test	94	106	71	778	1,049

このコンペティションは 2019 年にも開催されたが，そこで使用されたデータセットは，Twitter と Reddit の 2 つのプラットフォーム上での議論から収集されたものであった．本研究では，3.3 節で述べたように，Twitter 固有の機能から得られる情報を使用するので，Twitter のみから収集された 2017 年のデータセットを利用する．

4.2 評価尺度

評価尺度には Accuracy と Macro-F1 を用いる．Accuracy を評価尺度の 1 つとして選択した理由は，Accuracy が評価尺度として用いられた RumourEval の結果と提案手法の結果を比較するためである．もう 1 つの評価尺度として Macro-F1 を選択した理由は，データセットのクラスの偏りのためである．表 2 に示されるように，このデータセットはクラス間で含まれる数に大きな偏りがあり，具体的には最も関心の低いクラスである “Comment” のクラスが全体の 6 割以上を占めている．そこで，最も出現するクラスラベル（“Comment”）に過度な影響を受けることなく全てのクラスを平等に重み付けする Macro-F1 が評価尺度に適していると考えた．

ただし，提案した各手法の効果を比較検証する際には，Macro-F1 を重視するものとする．これは，上述したように本研究で扱うデータセットには大きな偏りがあるため，その偏りに影響を受けにくい Macro-F1 の値を軸としてモデルを構築することで，より良いモデルに近づくと判断したためである．実際に，2017 年ではなく 2019 年に開催された RumourEval においては，Macro-F1 が評価尺度として採用されていることから，Macro-F1 を判断の軸とするのは適切であると考えられる．

4.3 実験環境

実験に用いた CUDA のバージョンは 11.5，Python のバージョンは 3.7.12，pytorch のバージョンは 1.12.1，transformers のバージョンは 4.18.0，scikit-learn のバージョンは 1.0.2，numpy のバージョンは 1.21.6 である．

5 実験と考察

5.1 自然言語処理モデルの比較

3.1 節で述べたように，種々の言語モデルを使って得られた出力のうちの [cls] トークンに対応する分散表現を分類し，モデル間の性能を比較した．実験結果を表 3 に示す．また，種々の言語モデルを使って得られた出力のうちのツイート全体の分散表現をカプセルネットワークを用いて分類し，モデル間の性能を比較した．実験結果を表 4 に示す．表中の各モデルの特徴は，2.2 節にて詳述している．

4.2 節で述べたように，Macro-F1 の値に注目すると，どちらの結果においても， Twitter-RoBERTa を使って分類したモデルの値が最も高い数値となり，BERTweet で分類したモデルの値が最も低くなった．この 2 つのモデルの違いは，Twitter-RoBERTa は事前学習で用いた文章の中に規範的な文章を含むのに対し，BERTweet は事前学習の段階でツイートを学習したのみで，規範的な文章を学習していないことである．この結果は，ツイートというノイズが含まれた文章でのみ事前学習を行なった言語モデルよりも，規範的な文法に則った通常の長さの文章を用いて正しい文法規則を学習した上で，ツイートを学習した言語モデルの方が，より頑健なモデルになるということを示している．この事実は，直感的にも理解し得るものである．すなわち，我々人間も，規範的な文法を理解し，

表 3 自然言語処理モデルの比較 - [cls] トークンに対応する分散表現を用いた分類．なお，表中の $F1_D$ は，“Deny” のクラスの Macro-F1 スコアを表している．これは，表 4 においても同様である．

モデル	Accuracy	Macro-F1	$F1_D$
BERT	0.793	0.467	0.00
RoBERTa	0.761	0.455	0.00
Twitter-RoBERTa	0.788	**0.476**	0.12
BERTweet	0.742	**0.213**	0.00

表 4 自然言語処理モデルの比較 - ツイート全体の分散表現を用いた分類．

モデル	Accuracy	Macro-F1	$F1_D$
BERT	0.782	0.450	0.00
RoBERTa	0.757	0.375	0.00
Twitter-RoBERTa	0.742	**0.468**	0.12
BERTweet	0.742	**0.213**	0.00

長文を解釈できる能力があるからこそ，ノイズが含まれた文章を扱うことができ，また，短文からその文章が示すスタンスを理解することができる．規範的な文法に則った文章で学習をした BERT と RoBERTa が性能的に Twitter-RoBERTa と BERTweet の間の値となっていることも，それを表していると考えられる．

また，表 3 と表 4 には，4.2 節で述べた評価尺度の他に，“Deny” の F1 スコアを示す “$F1_D$” を示した．これに着目すると，Twitter-RoBERTa だけが “Deny” を正しく分類できていることが読み取れる．噂に関する投稿のスタンス分類のタスクでは，“Deny” は，そのラベルのデータ数の少なさなどの理由から，ほとんどが “Comment” に誤分類されてしまうというケースが多く，それが問題視されている．通常の文章を用いて正しい文法規則を学習した上で，ツイートというノイズが含まれた文章を学習した言語モデルを使うことで “Deny” の分類精度が上がり，それが Macro-F1 の値の向上に繋がったのは，着目すべき点である．

5.2 ターゲットツイートに関連するツイートの利用

3.2 節で述べたように，スタンス分類のターゲットとなるツイート (以降，Target Tweet，“TT” と略記する) 以外にも，TT に関連するツイートである，ソースツイート (以降，Source Tweet，“ST” と略記する)・TT の親ツイート (以降，Previous Tweet，“PT” と略記する) を利用することによる精度の変化を確かめた．なお，分類には，5.1 節にて有効性が確認された言語モデルである，Twitter-RoBERTa を用いた．[cls] トークンに対応する分散表現を分類することで比較した実験結果を表 5 に，ツイート全体の分散表現を用いてカプセルネットワークで分類することで比較した実験結果を表 6 に示す．

4.2 節で述べたように，Macro-F1 の値に注目すると，どちらの結果においても，関連ツイートを利用した分類の方が高い精度を得た．これは，ST や PT の情報が，スタンス分類にとっ

表 5　TT に関連するツイートの利用 - [cls] トークンに対応する分散表現を使った分類.

モデル	Accuracy	Macro-F1
TT のみ	0.788	0.476
ST と PT を利用	0.777	**0.491**

表 6　TT に関連するツイートの利用 - ツイート全体の分散表現を用いた分類.

モデル	Accuracy	Macro-F1
TT のみ	0.742	0.468
ST と PT を利用	0.744	**0.479**

て有用な情報を提供することを表す.

[SEP] トークンを使ってターゲットの文章以外の情報を与える方法は，実際に BERT を用いた自然言語処理タスクにおいて，文章に付随するトピックやタイトルなどの情報を同時に扱いたい時に利用する方法であり，あらゆるタスクにおいてその手法による精度の向上が確認されている．今回のケースでは，ST が，トピックやタイトルと一致するとまではいかなくとも，TT と関連のある情報を持っているために，精度の向上に繋がったと考察される.

また，TT と PT のスタンスの関連性を調べるために，TT のスタンスに対する PT のスタンスの割合を調べた．なお，ST の親となるツイートは存在せず，また，データセットの定義にて ST のスタンスは “Support” とされているため，データセットから ST と親が ST となるツイートを除いた全データを対象とした．その結果を，図 6 に示す．TT の各スタンスにおいて，PT として 1 番多いラベルは図中の緑色で表された “Comment” である．差分に注目すると，TT のどのスタンスにおいても大差なく現れているため，“Comment” はどのクラスのツイートに対しても続き得るスタンスだと言える．赤色の “Deny” に着目すると，TT のスタンスが “Deny” の時に PT が “Deny” である割合が，他のクラスと比べて高いことが読み取れる．実際にソーシャルメディア上で見かけることの多い，否定意見に対して否定意見を重ねる状況から，この事実は直感的にも理解される．このように，TT の持つスタンスと PT の持つスタンスには関連性があるので，PT の情報を分類で利用したことが精度の向上に繋がったと考察される.

5.3　Twitter 固有の機能から得られる情報の利用

3.3 節で述べたように，Twitter にはプラットフォーム固有の機能が存在する．5.1 節と 5.2 節にて有効性が確認された手法を利用をしたモデルをベースとして，Twitter 固有の機能から得られる情報の有効性を確かめる実験を行なった．その結果を，表 7 に示す．なお，表中の各モデルは，以下の特徴を持つ.

- Nothing：追加の特徴を利用しない.
- RT・Fav：RT 数といいね数を追加の特徴として利用.
- FF：FF 数を追加の特徴として利用.
- All：RT 数，いいね数，FF 数を追加の特徴として利用.

4.2 節で述べたように，Macro-F1 の値に注目すると，Twitter

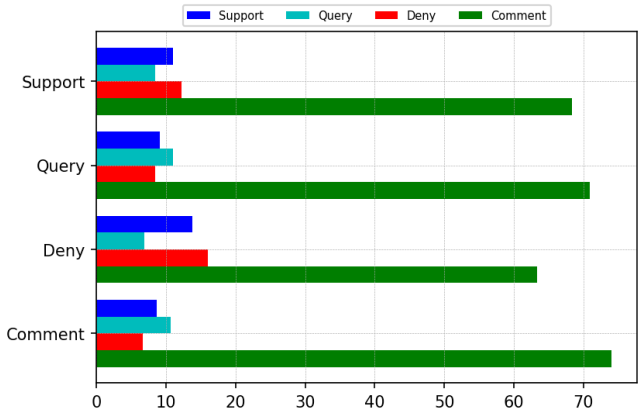

図 6　PT のスタンスの割合を，TT のスタンスごとに表示したグラフ．縦軸が TT のスタンスを表す．例えば赤色の “Deny” に着目すると，“Deny” は他のクラスに比べて “Deny” の後に連続して続きやすいという傾向が読み取れる.

表 7　プラットフォーム固有の特徴を利用した分類.

モデル	Accuracy	Macro-F1
Nothing	0.777	0.491
RT・Fav	0.746	0.506
FF	0.756	0.514
All	0.773	**0.528**

固有の特徴を利用することは効果的であることが分かる．追加の特徴を利用しなかった Nothing のモデルと，全ての追加の特徴を利用した All のモデルの Macro-F1 の差は歴然となった.

「RT」という行為は，RT するユーザーがツイートに何らかの影響を受け，そのツイートを拡散したいと思うことから始まり，「いいね」という行為は，いいねをするユーザーがツイートに対して同意することから始まる．いずれにせよ，RT やいいねをされるツイートは，他のユーザーにとって，そのスタンスが比較的明白なものであるということである．そのように，RT・いいね機能とツイートのスタンスの間に関連があることから，それらの特徴を分類に利用することが，精度の向上に繋がったと考察される.

FF 数を特徴として分類に利用することがスタンス分類の精度向上に繋がった理由を調べるために，FF 数とスタンスの関連を調べる．初めに，データセットに前処理を加え，調査の対象とするツイートとユーザーを限定する．データセットの定義上，ソースツイートは必ず “Support” が付与されるので，対象をソースツイート以外，すなわち，リプライツイートに限定した．データセットに含まれるツイートをしたユーザーの FF 数の最大値はそれぞれ 109,492 と 22,720,010 であったのに対し，中央値はそれぞれ 503.0 と 464.5 であることからも分かるように，このデータセットに含まれるツイートをしたユーザーの中には，その FF 数が全体と比べて大きすぎる，外れ値をとるユーザーが一定数存在した．そこで，グラフの見やすさを考え，各スタンスにおいて FF 数が上位 20 % の範囲に含まれるユーザーを除いたものを，調査の対象とした.

表 8　各スタンスにおけるフォロイー数・フォロワー数の関係を回帰分析した際の，回帰直線の傾きと決定係数．並びに，各スタンスを持つツイートをしたユーザーの中から，フォロイー数・フォロワー数上位 20% のユーザーを除いた中での，最大フォロイー数と最大フォロワー数．

	Support	Query	Deny	Comment
回帰直線の傾き	0.7627	**0.5962**	0.6661	0.6879
決定係数	0.4124	0.4278	**0.3634**	0.4662
最大フォロイー数	**1,819**	1.554	1.445	1,501
最大フォロワー数	**2,254**	1,472	1,666	1,513

各スタンスにおいて回帰分析をし，得られた回帰直線の傾きの数値と決定係数，さらに，調査の対象としたユーザーの各スタンスにおける最大の FF 数をまとめたものを，表 8 に示す．表における回帰直線の傾きを見ると，全体的に 1 を下回っていることから，調査の対象としたユーザーの FF 数は，フォロイー数の方が多い傾向にあることが分かる．ただ，その中でも “Query” が最も低い数値となっている．“Query” における FF 数の関係を，図 7 に示す．表 8 中の数値と図 7 から，リプライで “Query” のスタンスを持つツイートをするユーザーは，他のスタンスに比べてフォロワー数よりフォロイー数の方が多い傾向があることを表している．この事実は，直感的には，ソーシャルメディア上で他ユーザーによく疑問を投げかけるユーザー (このユーザーを A とする) が質問対象のユーザーをフォローすることはあっても，ユーザー A 自身が有益な情報を発信することは少ないので，あまりフォローされないというように解釈することができる．次に，決定係数に着目すると，“Support” ・“Query” ・“Comment” の決定係数はさほど変わらなかったが，“Deny” の決定係数は低い値を示した．これは，“Deny” のスタンスを持つリプライは，フォロイー数とフォロワー数の差分にあまり関係しないことを表す．次に，表 8 中の最大 FF 数に着目すると，“Query” ・“Deny” ・“Comment” では大体 1,500 ほどの数値を取っているのに対し，“Support” は 1,800 〜 2,300 ほどの高い数値となっている．このことから，“Support” のスタンスを持つリプライをするユーザーは，FF 数が多い傾向にあることが分かる．何らかの意見を支持し合う人間は，同じ考えを持った人間と集まりやすいという人間の特性から，このような結果が得られたと考察される．このように，ユーザーの FF 数はそのユーザーがするツイートのスタンスと何らかの関連があるため，FF 数を追加の特徴としてスタンス分類に利用することが，精度の向上に繋がったと考察される．

5.4　アンサンブル学習 (スタッキング) の実施

3.4 節で述べたように，言語モデルから得られた分散表現を余すことなく用い，また，本章の前節までの種々の実験によってその有効性が確認された手法を施したモデルを用いて，スタッキングの手法でアンサンブル学習を行う．有効性が確認された手法は，Twitter-RoBERTa の利用，関連ツイートの利用，Twitter 固有の機能から得られる数値の利用である．最終的なモデルの全体像は，3.4 節で示した図 5 のようになる．なお，

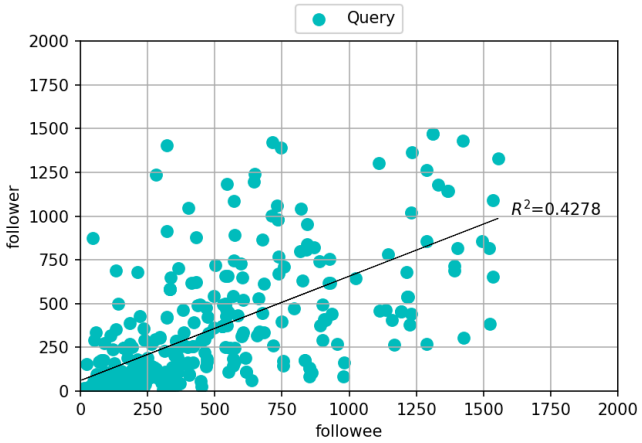

図 7　“Query” のスタンスを持つツイートをしたユーザーのフォロイー数とフォロワー数の関係．図中の直線は回帰直線，右上の数値は決定係数を表す．なお，外れ値の影響を受けないために，“Query” のスタンスを持つツイートをしたユーザーの中から，フォロイー数・フォロワー数上位 20% のユーザーを除いたものを表示対象としている．

本節は最終的に完成したモデルを RumourEval のコンペティション結果と比較する目的があるので，ベースラインとして以下のものを用いた．

- UWaterloo [10]：ツイートからトピックに依存しない特徴を抽出し，それを利用するモデル．テキストから得られる追加の特徴を利用するわけではないが，本研究においても 3.3 節で示したような追加の特徴を用いるので，UWaterloo をベースラインに採用した．このモデルは RumourEval で 2 番目に優れた精度を達成した．
- Turing [11]： Twitter 上での会話の木構造を用いた LSTM (Long Short-Term Memory) [12] に基づくスタンス予測をするモデル．本研究でもターゲットツイートと木構造上関連のあるツイートを利用した特徴抽出を行うので，Turing をベースラインに採用した．このモデルは RumourEval で最も優れた精度を達成した．

表 9 は，これらのモデルによって得られた結果を比較したものである．結果は， Accuracy，Macro-F1 ともに，本研究の提案モデルが最も高い精度を達成した．Turing が RumourEval において最も高いスコアであったことから，本研究の提案モデルの数値は，RumourEval のコンペティションで得られたものと比較して最も高い精度となる．アンサンブル学習手法の 1 つであるスタッキングと精度の向上の関係を確認する．アンサンブル学習を構成するそれぞれの分類器と本研究のモデルが示した Accuracy をまとめたものを，表 10 に示す．この表によれば，本研究が提案したモデルが最も高い Accuracy となっている．これは，3.4 節で述べたような，スタッキングという手法の，単体モデルよりも精度の向上に繋がりやすいという特徴と合致している．複数の分類器を用いることで各分類器の長所を使うことができたと考えられる．

表 9　ベースラインと，本研究が提案したスタッキングによるアンサンブル学習をするモデルの結果.

モデル	Accuracy	MacroF1
UWaterloo [10]	0.780	0.450
Turing [11]	0.784	0.434
本研究のモデル	**0.791**	**0.465**

表 10　アンサンブル学習を構成する種々の古典的な分類器の Accuracy と本研究のモデルの Accuracy をまとめた結果.

モデル	Accuracy
ロジスティック回帰	0.750
SVM	0.742
ランダムフォレスト	0.758
k 近傍法	0.747
勾配ブースティング	0.765
CapsNet	0.743
cls-feature	0.767
本研究のモデル	**0.791**

6　結論と今後の展望

本研究では，Twitter 上の噂に関する投稿のスタンス分類における既存研究がテキストデータを十分に扱えていないという問題点を指摘し，Twitter に特化した言語モデルを利用，ならびに，言語モデルの出力のうちの [cls] トークンに対応する分散表現を分類に使うだけではなく，ツイート全体の分散表現を分類に用いることを提案した．また，精度向上に繋がる手法として，分類のターゲットであるツイートと関連したツイートの利用，Twitter 固有の機能から得られる情報の利用を提案した．それらの手法の有効性を確認する実験を行った後に，各提案手法を組み込んだモデルを構築し，スタッキングの手法でアンサンブル学習を行ったところ，本研究で作成したモデルは，データセットが用いられたコンペティションでの結果と比較して最も高い精度を達成した

今後は，データセットのラベル間の数の偏りに対処していきたい．噂に関する投稿のスタンス分類では，表 2 で示したような不均衡データを扱うことが多い．こうしたデータは，一見 Accuracy が高くともそれが有用な結果とは言えないという事態や，データ数の少ないクラスの分類が難しいなどの問題が起こる．不均衡データ学習に用いられる数々のアプローチの中から噂のスタンス分類のタスクに適した手法を見つけ，この問題を解決していきたい．

文　献

[1] Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. Detection and Resolution of Rumours in Social Media: A Survey. *ACM Computing Surveys (CSUR)*, Vol. 51, No. 2, pp. 32:1–32:36, 2018.

[2] William Ferreira and Andreas Vlachos. Emergent: a novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pp. 1163–1168, 2016.

[3] Omar Enayet and Samhaa R El-Beltagy. NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 470–474, 2017.

[4] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic Routing Between Capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 3856–3866, 2017.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pp. 4171–4186, 2019.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, Vol. abs/1907.11692, , 2019.

[7] Francesco Barbieri, José Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Findings of the Association for Computational Linguistics (Findings of ACL: EMNLP 2020)*, pp. 1644–1650, 2020.

[8] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 9–14, 2020.

[9] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 69–76, 2017.

[10] Hareesh Bahuleyan and Olga Vechtomova. UWaterloo at SemEval-2017 Task 8: Detecting Stance towards Rumours with Topic Independent Features. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 461–464, 2017.

[11] Elena Kochkina, Maria Liakata, and Isabelle Augenstein. Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 475–480, 2017.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.