帰納バイアスを考慮したTwitter上での噂に関する投稿のスタンス検出

高田　大輔†　　杉山　一成†　　吉川　正俊†

† 京都大学情報学研究科　〒606–8501 京都市左京区吉田本町

E-mail: †daisuke.takata9@gmail.com, ††{kaz.sugiyama,yoshikawa}@i.kyoto-u.ac.jp

あらまし　今日，ソーシャルメディアプラットフォームは情報収集のツールとして多くの人々に利用されている．ソーシャルメディアプラットフォームの中でも，特に，Twitterは「ツイート」と呼ばれる短いメッセージをユーザーが自由に公開することができ，情報を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．しかし，ソーシャルメディアが膨大な情報へのアクセスを可能にする一方で，投稿された内容によってもたらされるソーシャルメディア上での風評被害や誤情報の伝播が，社会的な問題になっている．この問題に対処するべく，数多くの噂の信憑性を分類する手法が提案されている．なお，ここで「噂」とはその信憑性が曖昧である情報を指すものとする．噂の信憑性には，その噂に関する議論に参加しているものの噂に対するスタンスと相関関係にあることがわかっているので，噂の真偽判定には，その噂のツイートに対する意見のスタンス (“Support”，“Deny”，“Query”，“Comment”) の情報を利用することができる．そのため,「噂に関する投稿のスタンス分類」は,「噂の真偽判定」にとって非常に重要な役割を果たす．本研究では,「噂に関する投稿のスタンス分類」のタスクに着目し，その精度の向上につながると思われるような，帰納バイアスを考慮した3種類の手法を提案する．ここで，帰納バイアスとは，特定の条件・状況が分かっている際に，精度の向上につながると考えられるような制限をかけることを意味するものとする．1つ目は，特定のラベルを持つデータを追加することである．これは，本研究に用いるデータセットの特徴を考慮した上での提案手法である．2つ目は，皮肉の意味を持った単語の特徴抽出への利用である．これは，ソーシャルメディア上での会話に特有なパターンを考慮した上での提案手法である．3つ目は，Twitter 固有の情報の特徴抽出への利用である．これは，Twitter というプラットフォームに固有の機能を考慮した上での提案手法である．このうちの3つ目の手法は，大きく性能の向上に繋がることとなった．最後に，SemEval - 2017 Task 7 : RumourEval の subtaskA のデータセットに対して，提案手法の中で効果的だと判断された手法を用いて，アンサンブル学習を行う．結果は，Accuracy が 0.785，macro-F1 が 0.593 となった． これは，SemEval - 2017 Task 7 : RumourEval の subtaskA の結果の中で，最も高いスコアである．なお，このコンペティションでは，Accuracy が評価尺度として用いられている．

キーワード　ツイッター， 噂，情報の正確性，ソーシャルメディア

1 はじめに

ソーシャルメディアの発展により，人々は多くの情報を各種プラットフォームから得られるようになった．その中でも Twitter は,「ツイート」と呼ばれる短い文章でユーザーが自分の意見や自分が知る情報を自由に投稿することができ，情報を発信したり，収集したりするためのプラットフォームとして人気を博している．しかし，その利便性ゆえに，Twitter 上での誤情報の伝播が大きな問題となっている．2020 年には，新型コロナウイルスに関する誤情報が各地で混乱を引き起こした.「トイレットペーパーがなくなる」という誤情報が拡散されたために起きたトイレットペーパーの買い占めによる社会の混乱 [1] や，花崗岩を入浴剤として使うとコロナに効くといった誤情報が出回った [2] のはまだ記憶に新しい．こうした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も，噂に関する数多くの研究が行われてきた [3].

ここで，噂の定義を確認しておく．噂の定義は，研究ごとに様々な定義が存在するが，本研究では，ソーシャルメディアへの投稿時にはまだその信憑性が明らかになっていない情報のことを指すものとする．

噂には大きく分けて 2 つのタイプが存在する．

(1)　あるイベントが起こったタイミングで新しく出現する噂

(2)　その真偽が不明なまま長期間にわたり議論されている噂

(1) は，一般的には今までに検出されたことのない噂であるため，これを早期に検出することは，誤情報の拡散による混乱を防ぐという点で重要である．(2) は，その真偽が不明なまま長期間にわたり議論されている噂である．(1) の新しく出現するタイプの噂とは異なり，このタイプの噂はオンラインに処理される必要はない．

次に，噂に関する研究の概要について述べる．噂に関する研究は，数多く行われているが，大きく分けて 4 つの目的があり，そのうちの 1 つ，もしくはいくつかの組み合わせに焦点を当てている．以下にその 4 つの目的を示す．

(1)　噂の検出

(2)　噂の追跡

(3)　噂に関する投稿のスタンス分類

(4)　噂の真偽判定

「(1) 噂の検出」は，一連の投稿からどの投稿が噂を広めている投稿なのかを判断するものである.「(1) 噂の検出」のためのコンポーネントは，新しく出現した噂を特定するときに用いられ，長期間存在する噂を扱う場合には必要ない．主なアプローチとしては，投稿のテキスト情報を直接利用したテキストベースのアプローチ，その投稿をしたユーザー自身の情報を利用したユーザー情報ベースのアプローチ，ツイートの広まり方に関する情報を利用した伝播ベースのアプローチが存在する．

「(2) 噂の追跡」は，ある投稿が噂の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，もしくは前述の噂検出のコンポーネントによって特定された) 後に，その噂に関する意見を論じている投稿を見つけ，その投稿に関するクラスタを生成する．

「(3) 噂に関する投稿のスタンス分類」は，対象となる噂に関連した各投稿が，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．“Support” (ユーザーがその噂を支持する意図を持っている，S)・“Deny” (ユーザーがその噂を否定する意図を持っている，D)・“Query” (ユーザーがその噂に対して疑問を持っている，Q)・“Comment” (ユーザーがその噂に対して中立的な意見を述べている，C) の 4 種類のスタンスがある．このコンポーネントによって，後続の噂の真偽判定が容易になる．これは，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 [4] [5] により，噂の信憑性はその議論に参加しているもの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からである．すなわち,「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) 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かを判断するものである．判定と一緒にその判定がどの程度信頼できるものなのかを表す信頼度を出力する研究もある．

本研究では，上述した 4 つの目的のうちの 1 つである「(3) 噂に関する投稿のスタンス分類」の精度向上につながるような帰納バイアスを見つけることを目的とする．

2 関 連 研 究

本論文では，SemEval - 2017 Task 7 : RumourEval [6] の subtaskA 並びに SemEval - 2019 Task 7 : RumourEval [7] の subtaskA にて用いられたデータセットを用いて，噂に関連する投稿のスタンス分類を行う．スタンス分類が subtask の 1 つにまで設定されるのは，噂の真偽を分析する上で重要な点は，ソーシャルメディア上の他のユーザーが，その噂をどのように見ているかを定義することだからである．噂になるようなソースの投稿とその噂を議論する会話のスレッドがインプットとして存在する場合，スタンス分類の目的は，会話のスレッドの各投稿に，その噂に対するユーザーのスタンスをラベル付けすることである．このタスクが有効に機能することで，追加の文脈と情報を提供することにより，噂の真偽分類のタスクをサポートすることができる．例えば，ある一つの噂に対して議論が展開し，そのうちの多数が “Support” だった場合，議論を展開

会話スレッドの例 1

u1 : 新型コロナウイルス感染症は一般的に飛沫感染，接触感染で感染する．(Support)

u2 : @u1 その通りです．(Support)

u3 : @u2 信頼できる機関での HP でも確認できました，正しいです．(Support)

u4 : @u1 ちなみに飛沫感染とは感染者の飛沫 (くしゃみ，咳，つばなど) と一緒にウイルスが放出され，他の方がそのウイルスを口や鼻などから吸い込んで感染することを言います．(Comment)

u5 : @u1 専門家もそう言っていました．(Support)

u6 : @u1 間違えています．空気感染も一般的です．(Deny)

会話スレッドの例 2

u1 : WHO が方向転換して「感染者の隔離は不要」と発表した．(Support)

u2 : @u1 間違っています．(Deny)

u3 : @u1 信頼できません．(Deny)

u4 : @u3 私も信頼できません．WHO は会見でそのようなことを言っていません．(Deny)

u5 : @u1 情報源はありますか ? (Query)

u6 : @u1 WHO は改めて声明を発表するべき．(Comment)

図 1 会話のスレッドの例．例 1 は議論のもととなっている噂 (u1) が真である場合，噂に関する (スレッド上の) 投稿 (u2〜u6) には “Support” が多いことも表している．例 2 は議論のもととなっている噂 (u1) が偽である場合，噂に関する (スレッド上の) 投稿には “Deny” (u2〜u4) や “Query” (u5) が多いことも表している．

したユーザーたちがその議論のもととなった噂が真であることを検証したと推測することができる．なお，噂とそれに付随する議論による会話のスレッドの例を図 1 に示す．ソースとなる投稿にもスタンスがついている点には注意が必要である．

UWaterloo [8] のモデルは，ツイートからトピックに依存しない特徴を抽出し，それを利用することに焦点を当てたモデルである．トピックに依存しない特徴には，キュー特徴とメッセージ固有の特徴の 2 種類が存在する．キュー特徴とは，ツイートの話題の種類を特定するための手がかりとなるような単語から得られる特徴である．話題の種類は 9 種類存在する．そのうちの 3 種類を表 1 に示す．話題を表す単語を特定した後は，その話題の種類から投稿のスタンスを決定する．例えば，Belief や Knowledge に該当する単語は，ツイートした者が支持を表明している，すなわち “Support” のスタンスをとっていると考えられる．このようにして，キュー特徴を噂に関する投稿のスタンス分類に利用する．メッセージ固有の特徴とは，ツイートのテキスト情報から得られる特徴である．例えば，句読点の有無や，ツイートの文字数などがこれにあたる．これらの特徴を抽出した上で，ブースティングによる分類 (XGBoost) を行う．このモデルは， Accuracy 0.780 を達成した．これは，SemEval -

表 1　話題とそれを特定するための手がかりとなるような単語の例

Topic (Feature)	Example Words
Belief	assume，believe，suspect，think，thought
Knowledge	confirm，definitely，admit
Doubt	wonder，allege，unsure，guess，speculate

2017 Task 7 : RumourEval の subtaskA の中で，2 番目に優れた値である．

Turing [9] のモデルは，Twitter 上での会話の枝構造を用いた LSTM に基づくスタンス予測である．LSTM ベースの逐次モデルを提案し，ツイートの会話構造をモデル化した．このモデルは，Accuracy 0.784 を達成した．これは，SemEval - 2017 Task 7 : RumourEval の subtaskA の中で，最も優れた値である．

3　提 案 手 法

本研究では精度の向上につながると考えられるような，帰納バイアスを考慮した手法を 3 つ提案し，それぞれの結果を考察する．ここで，帰納バイアスとは，特定の条件・状況が分かっている際に，精度の向上につながると考えられるような制限をかけることを意味するものとする．3.1 節では，分類器の性能を改善させるために効果的だと考えられるデータセットについて考察する．これは，本研究に用いるデータセットの特徴を考慮した上での提案手法である．3.2 節では，word2vec アルゴリズム [1]を用いて抽出した特徴に続く形で皮肉の意味を持つ単語の特徴をより捉えるよう工夫することが，分類の性能を高めることにつながるかについて考察する．これは，ソーシャルメディア上での会話に特有なパターンを考慮した上での提案手法である．3.3 節では，Twitter 固有の情報を特徴抽出に利用することが分類の性能を高めることにつながるかについて考察する．これは，Twitter というプラットフォームに固有の機能を考慮した上での提案手法である．

最後に，SemEval - 2017 Task 7 : RumourEval の subtaskA でのデータセットに対して，提案手法のなかで効果的だと判断した手法を用いてアンサンブル学習を行い，SemEval - 2017 Task 7 : RumourEval の subtaskA での結果と比較する．

3.1　特定のラベルを持つデータの追加

本研究で用いる SemEval - 2017 Task 7 : RumourEval の subtaskA での訓練データセットには，“Deny” と “Query” のサンプル数が少ないという問題がある．そこで，ラスベガス銃乱射事件 (2017) とカリフォルニア銃乱射事件 (2018) の 2 つの事件に関連するツイートのうちの，“Deny” と “Query” に当てはまる 60 件のツイートをデータセットに加える．なお，ここで追加した分のデータセットは Aditya の研究 [10] で公開されているものである．これを，表 2 の訓練データセットで訓練した分類器，表 3 の訓練データセットで訓練した分類器の性能と比較し，考察する．テストデータセットは，SemEval - 2017 Task 7 : RumourEval の subtaskA のものを用いた．なお，データセットの詳細については，4.1 節で詳しく述べる．

u1 : ○○をすると絶対に△△に感染する．間違いない．(Support)
u2 : @u1 絶対という言葉の意味をご存知ですか？ (Deny)
u3 : @u1 ソースはどこですか？ (Query)

図 2　「?」を使っているが，仕様の用途が違う例．B さんの投稿では「?」は皮肉の意味で使われている．すなわち，絶対という言葉の意味を，相手が確実に知っているとわかった上でこのような聞き方をしているということであり，正しいラベルは “Deny” である．C さんは，質問の意図で「?」を用いているので，正しいラベルは “Query” である．

3.2　皮肉の意味を持った単語の特徴抽出への利用

ソーシャルメディア上での会話に特有なパターンがいくつかある [11]．その一つに，図 2 に示されるように，皮肉やレトリックなどを表現するときに「?」が含まれるというものがある．このパターンの問題は，「?」に引っ張られてしまい，誤って “Query” というラベル付けがされてしまう可能性があるという点である．この問題に対処するために，英語で皮肉を表す 41 語をリストとして保持し，それに含まれる語を特徴として捉えるようにする．4.2 節で述べるように，本研究において，データセットからの特徴抽出のうちの word2vec アルゴリズムを使ったモデルでは，悪口を表す英単語に関しては，英語で悪口を表す語 [2]をリストとして保持し，それに含まれる語を特徴として捉えるようにしている．これは，2 章で述べた，Turing のモデルを参考にしてのことである．そこで，以下の 4 パターンの性能の変化を比較して，考察する．

(1)　悪口も皮肉も特徴としないモデル
(2)　悪口を特徴とするモデル
(3)　皮肉を特徴とするモデル
(4)　悪口と皮肉の両方を特徴とするモデル

3.3　Twitter 固有の情報の特徴抽出への利用

これまでの，投稿からのスタンス検出の際の特徴抽出では，投稿のテキスト情報を直接利用した，テキストベースでの特徴抽出が多かった．例として，「?」の有無を特徴として捉えたり，写真やビデオなどのメディアの有無を特徴として捉えたりするものがある．4.2 節で述べる特徴抽出の方法は，全てテキストベースの特徴抽出である．また，3.2 節で提案した手法も，テキストベースの特徴抽出であると言える．

ここで，Twitter というプラットフォーム自体に着目すると，このプラットフォームには，プラットフォーム特有の数々の機能がある．代表的なものに，以下の 3 つがある．

(1)　フォロー機能
(2)　リツイート機能
(3)　お気に入り機能

1 : https://code.google.com/archive/p/word2vec/

2 : http://urbanoalvarez.es/blog/2008/04/04/bad-words-list/

表 2　SemEval - 2017 のデータセット

	Support	Deny	Query	Comment	Total
train	841	333	330	2,734	4,238
test	94	71	106	778	1,049

表 3　SemEval - 2019 のデータセット

	Support	Deny	Query	Comment	Total
train	910	344	358	2,907	4,519
test	141	92	62	771	1,066

1 章での，噂の研究の概要で述べた「(1) 噂の検出」には，その投稿をしたユーザー自身の情報を利用したユーザー情報ベースのアプローチが存在する．したがって，噂に関する投稿のスタンス検出にもユーザの情報を利用できると考えられるので，フォロー数・フォロワー数を特徴として捉える．また，リツイート数やお気に入り数が多いツイートは，それを見る側のユーザーにとって受け止めやすいもの，すなわち，スタンスが比較的明瞭なものであると考え，それら 2 つの指標も特徴として捉える．

最後に，SemEval - 2017 Task 7 : RumourEval の subtaskA でのデータセットに対して，提案手法のなかで効果的だと判断した手法を用いて，ハイパーパラメータチューニングをした分類器でアンサンブル学習を行う．この学習法は，様々な分類アルゴリズムを組み合わせることで，個々のモデルの弱点を補完するので，分類器を個別に使う方法よりも高い汎化性能が得られる．それを，同じデータセットを用いている SemEval - 2017 Task 7 : RumourEval の subtaskA での結果と比較し，考察する．なお，ここで SemEval - 2019 Task 7 : RumourEval の subtaskA を選択しなかったのは，2019 年のコンペティションのデータセットには Reddit のデータが含まれており，3.3 節での Twitter 固有の情報を利用した特徴抽出が実現できないためである．また，Twitter と Reddit で出現する噂のタイプが異なるためでもある．これについては 5 章で詳しく述べる．本研究では，あくまでも Twitter 上のデータに対するスタンス検出という課題に焦点を当てる．

4 実　　験

4.1 データセット

本研究では，2 種類のデータセットを用いて実験を行う．

表 2 は，SemEval - 2017 Task 7 : RumourEval の subtaskA でのデータセットである．

表 3 は，SemEval - 2019 Task 7 : RumourEval の subtaskA でのデータセットのうちの Twitter のデータセットである．Twitter のデータセットに限定しているのは，もともとこのデータセットは Twitter と Reddit の 2 つのソーシャルメディアプラットフォーム上での議論から収集されたものであることと，3.3 節で述べたように，本研究ではあくまでも Twitter 上のデータに対するスタンス検出という課題に焦点を当てるからである．

これらのデータセットは，どちらも Twitter 上での議論から収集されたものである. そのトピックは，そのデータセットが作られた当時に話題となったトピックを扱っている．そのスタンスのアノテーションには，クラウドソーシングが用いられている．アノテーションには，まず，参加できる人を米国と英国の人々に限定させ，各アノテーションを最大 10 人の担当者で行い，担当者の間での一致度が 70%である時に正しいラベル付けがされたとしている．

4.2 特 徴 抽 出

テキストデータからの特徴抽出は，Hareesh の研究 [8] と Aditya の研究 [10] を参考に，word2vec アルゴリズムを使ったモデルを実装した．

はじめに前処理として，文書のトークン化とストップワードの除去を行う．ストップワードの除去には， NLTK ライブラリ [3]で提供されている 127 個の英語のストップワードを指定した．

Google News のデータセットで事前学習済みのモデルを利用して，ツイートに対する特徴ベクトルを生成する．word2vec アルゴリズム [12] は，ニューラルネットワークに基づく教師なし学習アルゴリズムであり，単語間の関係を自動的に学習しようとする．背景には，同じような意味を持つ単語を同じようなクラスタに配置するという考え方がある．また，これに続く形の特徴ベクトルとして，2 章で述べた Uwaterloo のモデルを参考に，次の 10 個の特徴を追加した．

- 「not」や「no」,「don't」などといった否定語の数.
- 句読点の数.
- 「?」の有無．疑問のスタンスを持つツイートは「?」を含むことが多い.
- 感嘆符の有無.
- ハッシュタグの有無.
- ユーザーへのメンションの有無.
- URL の有無.
- 写真やビデオなどのメディアの有無.
- テキスト処理に使うことのできるライブラリである Textblob[4] を用いた感情分析 (極性分析)．本研究では，結果が肯定的なものであれば 1，そうでなければ 0 とした.
- 事前に登録された 458 個の悪口の表現の中に含まれる語数．否定のスタンスを持つツイートは悪口表現を含むことが多い.

ベースライン作成時の特徴抽出や，提案手法の実験での特徴抽出の際などにこのモデルを用いる時は，全てこれら 10 個の追加の特徴ベクトルを追加したものとする．

4.3 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 つ目として選択した理由は，SemEval - 2017 Task 7 : RumourEval の subtaskA において，

3 : https://www.nltk.org/

4 : https://textblob.readthedocs.io/en/dev/

それが評価尺度として用いられたためである．4.1 節で述べたように，今回の実験で使用するデータセットの 2 種類のうちの一つは，SemEval - 2019 Task 7 : RumourEval の subtaskA のものである．提案手法の結果をこのタスクの結果と比較する時のために Accuracy を評価尺度の 1 つ目として選択する．

macro-F1 を評価尺度の 2 つ目として選択した理由は，SemEval - 2019 Task 7 : RumourEval の subtaskA での評価尺度が macro-F1 であることと，データセットのクラスの偏りのためである．前者は，accuracy を選択した理由と同じ理由である．後者は，表 2 と表 3 に示されるように，このデータセットは含まれるクラスに大きく偏りがあり，具体的には最も関心の低いクラスであると言えるコメントのクラスが，訓練データ・テストデータともに 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 (すなわちコメントのクラス) に過度な影響を受けることなく分類器の全体の性能を評価するよう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めである．

4.4 ベースラインの作成

3 章で述べた各手法の効果を検証するにあたって，比較するためのベースラインが必要であるが，4.1 節で述べたように，今回の実験で用いるデータセットは，公式のコンペティションで用いられているデータセットだけでなく，それに対して自身で改良を加えたものも含まれるので，コンペティションでのベースラインは本研究のベースラインとしては相応しくない．そこで，自身でベースラインを作成する．

4.4.1 ベースラインのデータセット

ベースラインに対して用いるデータセットとしては，4.1 節で述べた 3 種類のデータセットのうち，一番データの数が少ない，SemEval - 2017 Task 7 : RumourEval でのデータセットを用いる．データセットとして一番データの数が少ないものを選んだのは，3.1 節にて，データを増やすことでの性能の変化を比較するためである．

なお，テキストからの特徴抽出としては，4.2 節で述べた方法を用いる．

4.4.2 ベースラインの候補の作成

作成した分類器を以下に示す．

(1) Naïve Bayes 分類器 (NB) : テキストの分類問題のための基本的な分類器である．他のアルゴリズムと比べて，比較的小さなデータセットで特に優れた分類性能を発揮する傾向があるので，ベースラインとして選択した．パラメータ α を大きくするとモデルの複雑さが減るが，アルゴリズムの性能はそれほど変化しないことが多い．今回はラプラススムージングに相当する．結果は，Accuracy が 0.726，macro-F1 が 0.248 となった．

(2) Support Vector Machine (SVM) : SVM はカーネルトリックを使って分離超平面を高次元空間で特定することができる．カーネルとして，動径基底関数カーネル (以降は RBF カーネルとする) を用いる．γ パラメータはこの RBF カーネルのカットオフパラメータであると解釈できる．このパラメータによって訓練データセットの影響力が直接決まるので，このパラメータを最適化することが，過学習の制御の面で重要な役割を果たす．そこで，γ，正則化の強さを決めるパラメータ C，乱数については，10 分割交差検定で，グリッドサーチとランダムサーチを用いて決定した．結果は，Accuracy が 0.777，macro-F1 が 0.561 となった．

(3) ロジスティック回帰 (LR) : 最適解の探索手法として L-BFGS アルゴリズムを用いたロジスティック回帰で実験を行う．多クラス分類の種類として，あるインスタンスを対象のクラスに割り当てるか，それ以外に割り当てるかという二値分類の拡張か，純粋な多クラス分類かを指定しなければならないが，今回は訓練データが複数のクラスに所属することはなく，1 つのクラスにしか所属しないことが保証されている互いに排他なデータなので，通常推奨されている多クラス分類を指定した．なお，クラス分類に二値分類の拡張を指定しなければ，最適解の探索手法の LIBLINEAR アルゴリズムを使うことはできないこともあり，L-BFGS アルゴリズムを指定している．正則化に L1 ノルムを用いるか L2 ノルムを用いるかということと，正則化の強さを決めるパラメータ C，乱数については，10 分割交差検定で，グリッドサーチとランダムサーチを用いて決定した．結果は，Accuracy が 0.776，macro-F1 が 0.566 となった．ロジスティック回帰は識別性が高いため，NB よりも良い結果を得ることができた．

(4) ランダムフォレスト (RF) : 決定木のアンサンブルと見なすことができるランダムフォレストは，スケーラビリティが高く，使いやすいことで知られている．これは，それぞれバリアンスが高い複数の決定木を平均化することで，より汎化性能が高く過学習に対して堅牢なモデルを構築するという考え方に基づいている．不純度の指標には，ジニ不純度を採用した．一般的にジニ不純度とエントロピーは，非常によく似た結果となる．決定木モデルの構築の際には，ブートストラップサンプリングを行うように指定した．ここで，ブートストラップ標本のサイズを適切に設定することが重要である．ブートストラップ標本のサイズを小さくすると，ランダムフォレストのランダム性が向上することがあるので，過学習の影響を抑えるのに役立つ可能性がある．ただ，その分バイアスは大きくなってしまう．逆にブートストラップ標本のサイズを大きくすると，過学習に陥る可能性が高くなる一方で，バリアンスが大きくなってしまう．学習結果に大きく影響するこのパラメータと乱数については，10 分割交差検定で，グリッドサーチやランダムサーチを用いて決定した．結果は，Accuracy が 0.770，macro-F1 が 0.623 となった．

(5) k 最近傍法 (KNN) : KNN アルゴリズムは選択された距離指標に基づき，訓練データセットの中から分類したいデータに最も近い k 個の訓練データを見つけ出す．そしてそのデータ点のクラスラベルは k 個の最近傍の多数決によって決まる．距離指標にはユークリッド距離を用いた．最近傍の点を何点選択するかは，過学習と学習不足のバランスをうまくとらなければならない．このパラメータは 10 分割交差検定で，グリッドサーチを用いて決定した．結果は，Accuracy が 0.757，macro-F1

表 4　最終的に決定したベースライン

分類器	Accuracy	macro - F1
NB	0.726	0.248
SVM	0.777	0.561
LR	0.776	0.566
RF	0.770	0.623
KNN	0.757	0.460
勾配ブースティング	0.776	0.566

が 0.460 となった．

(6)　ブースティング : ブースティングでは，誤分類された訓練データを後から弱学習器に学習させることで，アンサンブルの性能を向上させる．今回はブースティングアルゴリズムのうちの勾配ブースティングに焦点を当てる．これも弱学習器を強学習器にブーストするものである．これには scikit-learn の GradientBoosting と Python のライブラリの API の XGBoost を採用した．XGBoost は，基本的には勾配ブースティングの計算効率の良い実装である．GradientBoosting での弱学習器の個数は性能に直接影響する．そこで，このパラメータはグリッドサーチを用いて決定した．勾配ブースティングの結果は，Accuracy が 0.776，macro-F1 が 0.566 となった．XGBoost の結果は，Accuracy が 0.765，macro-F1 が 0.514 となった．

4.4.3　ベースラインの決定

表 4 は，最終的にベースラインとして決定した分類器の Accuracy と macro-F1 をまとめたものである．

4.5　実 験 結 果

3 章で述べた各手法に対する実験結果を述べ，その結果を考察する．

4.6　特定のラベルを持つデータの追加

SemEval - 2017 Task 7 : RumourEval の subtaskA のデータセットを 2017data，SemEval - 2017 Task 7 : RumourEval の subtaskA のデータセットに “Deny” と “Quary” に当てはまるデータセットを足したものを 2017v2data，SemEval - 2019 Task 7 : RumourEval の subtaskA のデータセットのうちの Twitter のデータセットを 2019data とする．これら 3 種類のデータセットに対して 4.4.2 節で述べた種々の分類器を用いて学習を行い，その性能の変化を考察する．なお，テストデータセットは，SemEval - 2017 Task 7 : RumourEval の subtaskA のテストデータセット，特徴抽出は 4.2 節の word2vec アルゴリズムを用いたモデルを用いた．実験結果を表 5，表 6 に示す．

ランダムフォレスト (RF) に着目すると，Accuracy があまり変化していないのに対して，macro-F1 は，大きく変動していることがわかる．具体的には，2017 に対して 2017v2 と 2019 のスコアが減少している．そこで，2017v2 と 2019 での，訓練データに対する Accuracy を求めると，前者は 0.995，後者は 0.995 と，明らかに過学習を起こしていた．ランダムフォレストでは，今のパラメータのままでは，これ以上データを増やしても過学習を起こすだけであると考えられる．

他の分類器の macro-F1 に着目すると，データセットの変更による性能の変化があまりないことがわかる．データの前処理の仕方，特徴抽出の方法，分類器自体の構成を工夫することが大事であると考えられる．

表 5　訓練データセットの変更による性能の変化 1

	NB		SVM		LR	
	acc	macroF1	acc	macroF1	acc	macroF1
2017	**0.726**	**0.248**	0.777	**0.561**	0.776	**0.566**
2017v2	**0.726**	**0.248**	**0.780**	0.561	**0.786**	0.558
2019	**0.726**	0.245	0.779	0.560	0.783	0.565

表 6　訓練データセットの変更による性能の変化 2

	RF		KNN		ブースティング	
	acc	macroF1	acc	macroF1	acc	macroF1
2017	**0.770**	**0.623**	0.757	0.460	0.776	0.566
2017v2	0.745	0.457	0.757	0.460	0.776	**0.582**
2019	0.765	0.516	**0.766**	**0.508**	**0.782**	0.570

4.7　皮肉の意味を持った単語の特徴抽出への利用

データセットは，SemEval - 2017 Task 7 : RumourEval の subtaskA のデータセットを用い，特徴抽出は 4.2 節の word2vec アルゴリズムを用いたモデルを用いた．表 7，表 8 に，実験結果を示す．

ロジスティック回帰 (LR)，ランダムフォレスト (RF)，ブースティングの nothing，bad の macro-F1 に着目すると，総じて値が増加している．ここで，ランダムフォレストの nothing，bad の時の混合行列を表 9，表 10 に表す．悪口を表す単語を特徴として捉えることで，全てのラベルにおいて，正しく分類することができたデータの数が増えていることがわかる．この中でも特に着目すべき点は，“Query”(Q) と “Deny”(D) において正しく分類できたデータの数が増えたことである．この二つのラベルは分類が難しく，大抵が “Comment”(C) として識別されてしまう．“Deny”(D) では F1 スコアが 0 となることも多い．この二つのラベルでの精度が向上したのは，“Query”(Q) や “Deny”(D) のスタンスを持つテキストに悪口を示す表現が含まれることがしばしばあるためであると考えられる．

次に，ロジスティック回帰 (LR)，ランダムフォレスト (RF)，ブースティングの nothing，bad & irony の macro-F1 に着目すると，総じて値が増加している．しかし，NB，ロジスティック回帰以外の分類器の bad，bad & irony の macro-F1 に着目すると，総じて値が減少しているのがわかる．ここから考えられることは，悪口の言葉の特徴と皮肉の言葉の特徴を捉えることは確かに精度の向上にはつながるが，二つとも足すことは特徴量が多過ぎて精度が落ちてしまうということである．

また，nothing と irony の macro-F1 に着目すると，ランダムフォレスト (RF) とブースティングに関しては精度の向上につながったが，それ以外では精度が向上することはなかった．皮肉の言葉の特徴を捉えることが精度の向上にあまり繋がらなかったのは，英語での皮肉表現が，文章の形で表されるものが

表 7　特定の意味を持つ語の特徴を捉えることによる性能の変化 1

	NB		SVM		LR	
	acc	macroF1	acc	macroF1	acc	macroF1
nothing	**0.726**	**0.248**	**0.777**	**0.561**	0.774	0.561
bad	**0.726**	**0.248**	**0.777**	**0.561**	0.776	0.566
irony	**0.726**	**0.248**	**0.777**	**0.561**	0.774	0.561
all	**0.726**	**0.248**	0.776	0.559	**0.777**	**0.570**

表 8　特定の意味を持つ語の特徴を捉えることによる性能の変化 2

	RF		KNN		ブースティング	
	acc	macroF1	acc	macroF1	acc	macroF1
nothing	0.748	0.460	0.755	**0.462**	0.774	0.549
bad	**0.770**	**0.623**	**0.757**	0.460	**0.776**	0.566
irony	0.760	0.569	0.754	0.455	**0.776**	**0.609**
all	0.766	0.611	0.756	0.459	0.770	0.550

表 9　nothing での混合行列

	S	Q	D	C
S	25	1	0	68
Q	1	14	0	91
D	1	0	0	70
C	18	12	2	746

表 10　bad での混合行列

	S	Q	D	C
S	26	0	0	68
Q	1	26	0	79
D	2	1	1	67
C	15	6	2	755

多いことが一因として考えられる．また，皮肉を表す単語として登録した語の数が少なかったことも一因として考えられる．

4.8　Twitter 固有の情報の特徴抽出への利用

データセットは，SemEval - 2017 Task 7 : RumourEval の subtaskA のデータセットを用い，特徴抽出の方法としては 4.2 節の word2vec アルゴリズムを使ったモデルを使用する．それによって抽出された特徴に続く形として以下のものを加える．なお，この名称は，結果を示す表 11，表 12 と対応している．

- base : 何も加えない，基準となる値．
- user : フォロワーの数，フォロイーの数 (ユーザー情報) を特徴として加える．
- ret & fav : リツイート数，お気に入り数を加える．
- all : ユーザー情報，リツイート数，お気に入り数を加える．

表 11，表 12 に，4.4 節で用意した各分類器によって得られた実験結果を示す．

SVM と k 最近傍法 (KNN) の macro-F1 に着目すると，user と all が base に比べて非常に低い値となっている．そこで，SVM の ret & fav での訓練データに対する Accuracy と all での訓練データに対する Accuracy を出力すると，前者は 0.752 だったのに対し，後者は 0.924 であった．後者は明らかに過学習を起こしていることがわかる．ユーザーに関する情報を加えることが特徴量を増やし過ぎてしまうことになり，それによって過学習が起きていることが考えられる．

表 11　Twitter 固有の情報を特徴抽出に利用することによる性能の変化 1

	NB		SVM		LR	
	acc	macroF1	acc	macroF1	acc	macroF1
base	**0.726**	0.248	0.777	**0.561**	0.776	0.566
user	0.127	0.235	0.741	0.185	0.756	0.424
ret & fav	0.090	0.022	**0.780**	0.545	**0.783**	0.580
all	0.127	**0.253**	0.742	0.185	0.753	**0.646**

表 12　Twitter 固有の情報を特徴抽出に利用することによる性能の変化 2

	RF		KNN		ブースティング	
	acc	macroF1	acc	macroF1	acc	macroF1
base	**0.770**	0.623	0.757	0.460	0.776	0.566
user	0.763	0.526	0.744	0.308	0.774	**0.601**
ret & fav	0.765	**0.759**	**0.780**	**0.548**	**0.779**	0.595
all	0.760	0.503	0.746	0.314	0.776	0.568

次に，ロジスティック回帰 (LR)，ランダムフォレスト (RF)，k 最近傍法 (KNN)，ブースティングの macro-F1 に着目すると，ret &fav が base に対して一様に高い数値を示している．特に，ランダムフォレストの macro-F1 は 0.759 と，他の分類器の値と比べても非常に高い値を示している．このことから，リツイート数・お気に入り数を特徴として利用することが性能向上につながると言える．これは，リツイート数・お気に入り数が多いツイートは，そのスタンスがユーザー側にとって比較的明瞭，すなわち，スタンスが分類しやすいことが関係していると考えられる．

4.9　効果的な帰納バイアスの検証

最後に，これまでの 4.5 節を通して効果的だと判断した帰納バイアスを用いて実験を行う．データセットには SemEval - 2017 Task 7 : RumourEval の subtaskA のデータセットを用い，このコンペティションの結果と比較する．データセットからの特徴抽出には，4.2 節の word2vec アルゴリズムを使ったモデルを使用する．4.7 節より，悪口を表す言葉の特徴を捉えるようにする．4.8 節より，リツイート数・お気に入り数を特徴として抽出する．それをハイパーパラメータチューニングをした上での SVM，ロジスティック回帰，ランダムフォレスト，k 最近傍法，勾配ブースティングを用いたアンサンブルで学習した結果を表 13 に示す．UWaterloo [8] のモデルの Accuracy は 0.780，Turing [9] のモデルの Accuracy は 0.784 なので，これは，SemEval - 2017 Task 7 : RumourEval の subtaskA の中で最も高い数値である．

表 13　効果的な帰納バイアスの検証

Accuracy	macro-F1
0.785	0.593

5 結　　論

本研究では，噂に関する投稿のスタンス検出に有効であると考えられる手法を 3 つ提案した．

1 つ目は，「特定のラベルを持つデータの追加」である．これは，スタンス検出に利用するデータにクラスの偏りがあることを考慮して，特にその数が少ない “Deny” と “Query” に相当するツイートを，追加のデータセットとして加えるものであった．ベースとなるデータセット，特定のクラスのラベルを持ったデータを追加したデータセット，特定のクラスでなく全体的にデータを追加したデータセットの 3 種類の訓練データセットを用意し，訓練後に得られたそれぞれの分類器の性能を比較したが，その性能に大きな差はなかった．

2 つ目は，「特定の意味を持った単語の特徴抽出」である．これは，“Deny” や “Query” の表現に含まれることの多い，悪口の表現や皮肉の表現を持つ単語を特徴抽出に利用することによって，性能の向上を図るものであった．悪口の表現や皮肉の表現を特徴抽出に利用することによって精度は向上したが，悪口の表現のみ特徴抽出に利用するほうが精度は向上の幅は大きかった．

3 つ目は，「Twitter 固有の情報の特徴抽出」である．これは，Twitter というプラットフォームに固有の情報を特徴抽出に利用するものである．Twitter 固有の情報には，大きく分けて，「ユーザー関連の情報」と「リツイート数・お気に入り数の情報」の 2 種類が存在する．ユーザー関連の情報を足すことは，特徴量が増えすぎることになってしまい，過学習に陥り．性能が低下した．しかし，テキスト関連の情報を足すことは，スタンス検出の精度の向上に繋がった．これは，リツイート数・お気に入り数が多いツイートは，そのスタンスがユーザー側にとって比較的明瞭，すなわち，スタンスが分類しやすいことが関係していると考えられる．

最後に，効果的であると判断された帰納バイアスを考慮して，SemEval - 2017 Task 7 : RumourEval の subtaskA のデータセットと同じデータセットで実験を行い，このコンペティションの結果と比較したところ，コンペティション結果の中で最も高い数値を得ることができた．

今後は，今回の研究で得られたことを利用して，Twitter とは異なるプラットフォームでの噂に関する投稿のスタンス検出にも取り組んでいきたい．しかし，プラットフォーム間での現れる噂の種類の違いがこの問題を難しくする．特に，Twitter と Reddit ではこのプラットフォーム間の違いが明白である．Twitter で出現する噂にはニュース速報的なものが多いのに対し，Reddit で出現する噂は長年議論されている噂であることが多い [11]．すなわち，1 章で述べた噂の二つのタイプのうちの「(1) あるイベントが起こったタイミングで新しく出現する噂」がよく現れるのが Twitter であり，「(2) その真偽が不明なまま長期間にわたり議論されている噂」がよく現れるのが Reddit であるということである．また，Twitter 上での噂に関する議論はそれが収束するまでに長い時間がかかるのに対し，Reddit 上での噂に関する議論は比較的短い時間で収束する [13]．SemEval - 2019 Task 7 : RumourEval の subtaskA では，この問題により分類の難易度が上がった．

文　　献

[1] 江口英佑, 栗林史子: 朝日新聞（2020 年 3 月 25 日）デマ拡散，トイレットペーパー消えた「在庫は十分」.

[2] 朝日新聞: 朝日新聞（2020 年 2 月 28 日） (新型コロナ) デマ・便乗商法に注意「予防効果」現時点では全て根拠なし.

[3] Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M. and Procter, R.: Detection and resolution of rumours in social media: A survey, *ACM Computing Surveys (CSUR)*, Vol. 51, No. 2, pp. 1–36 (2018).

[4] Ferreira, W. and Vlachos, A.: Emergent: a novel data-set for stance classification, *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 1163–1168 (2016).

[5] Enayet, O. and El-Beltagy, S. R.: NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter., *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pp. 470–474 (2017).

[6] Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Wong Sak Hoi, G. and Zubiaga, A.: SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 69–76 (2017).

[7] Gorrell, G., Kochkina, E., Liakata, M., Aker, A., Zubiaga, A., Bontcheva, K. and Derczynski, L.: SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours, *Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 845–854 (2019).

[8] Bahuleyan, H. and Vechtomova, O.: UWaterloo at SemEval-2017 Task 8: Detecting stance towards rumours with topic independent features, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 461–464 (2017).

[9] Kochkina, E., Liakata, M. and Augenstein, I.: Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm, *arXiv preprint arXiv:1704.07221* (2017).

[10] Adhikary, A., Singal, S. and Kapur, S.: Rumour Stance Classification: NLP Final Project Report (2019). NLP Final Project Report.

[11] Radhakrishnan, K., Kanakagiri, T., Chakravarthy, S. and Balachandran, V.: “A Little Birdie Told Me...”-Social Media Rumor Detection, *Proceedings of the 6th Workshop on Noisy User-generated Text (W-NUT 2020)*, pp. 244–248 (2020).

[12] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient estimation of word representations in vector space, *Proceedings of the Workshop at the 1st International Conference on Learning Representations(ICLR 2013)* (2013).

[13] Priya, S., Sequeira, R., Chandra, J. and Dandapat, S. K.: Where should one get news updates: Twitter or Reddit, *Online Social Networks and Media*, Vol. 9, pp. 17–29 (2019).