BERT を利用した煽りツイート検出の一手法

松本 典久† 上野 史‡ 太田 学‡

†岡山大学工学部情報系学科 〒700-8530 岡山県岡山市北区津島中 3-1-1
‡岡山大学大学院自然科学研究科 〒700-8530 岡山県岡山市北区津島中 3-1-1
E-mail: †piia6ope@s.okayama-u.ac.jp, ‡{uwano, ohta}@okayama-u.ac.jp

あらまし 近年、パソコンやスマートフォンの普及に伴い SNS が広く用いられるようになる一方で、炎上に代表される SNS におけるネットトラブルが問題となっている．それを誘引する一因が煽り投稿だが，その表現は多岐にわたるため，表現が直接的に文面に現れないものも存在し，精度の高い検出を困難にしている．本研究では，Twitter の投稿であるツイートを対象として，前後の文脈を反映できる Bidirectional Encoder Representations from Transformers (BERT) を用いた煽りツイートの検出方法を提案する．検出対象とするツイートは煽り行為をしているリプライツイートとし，BERT の分類器を作成して，ツイートが煽りか否かを二値分類することで煽りツイートを検出する．実験の結果，非線形 SVM などの分類器に比べて BERT の分類器は高い精度で検出でき，BERT を利用することの有効性が確認された．また，ツイートのリプライ元と合わせたデータであるツイート組の利用が，同一文面で煽りと非煽りが異なるリプライツイートの分類に有効であることが確認された．

キーワード Twitter，煽りツイート，リプライツイート，機械学習

1. はじめに

近年，パソコンやスマートフォンの普及に伴い Twitter や Facebook，Instagram などの SNS が広く用いられるようになり，多くの人が自分の考えをネット上で容易に発信できるようになった．一方で，それに伴い「ネット炎上」に代表されるネットトラブルが問題となっている．ネットトラブルは企業や業界人，一般人など問わず甚大な被害を与え，収束のために謝罪文の掲載や担当者の解雇などの対応に追われることもあれば，当事者の自殺などにまで発展することもある．それらの一因が煽り投稿という「相手の感情を逆撫でる内容の投稿」であり，これは相手から失言を引き出すことや関係のない他者の注目を集めることなどを目的としている．そのため，トラブルを発生，あるいは拡大させる元凶ともなり，SNS などではその検出を実施しているものもある．しかし，煽り投稿は表現が多岐にわたり，文面に直接的に現れないものも存在するため検出は難しい．

本研究では Twitter の投稿であるツイートを対象として煽り投稿であるツイート，すなわち煽りツイートを検出する．検出対象をリプライツイートとし，文脈などを考慮する分類器を作成して対象のツイートが煽りか否かを分類することで煽りツイートを検出する．また検出には自然言語処理モデルの一つである Bidirectional Encoder Representations from Transformers (BERT) [1] を利用する．BERT の特徴に文脈を読むことが可能となったことが挙げられる．これにより，対象となるリプライツイート内の文脈を学習させることで，直接的な表現として現れない煽りを正しく判定することが期待される．また，本研究ではリプライツイートの分類器のほかに，同一文面のリプライツイートを文脈に応じて正しく判定することを期待して，そのリプライ元と合わせたデータを分類対象とした分類器を提案し，有効性を比較する．

本論文の構成を述べる．第 2 節では，関連研究を紹介する．第 3 節では，BERT の概要と提案手法である BERT を利用した分類器について説明する．第 4 節で，評価実験として煽りツイートの分類実験の内容と結果を示し，第 5 節で考察を述べる．第 6 節で，本論文のまとめと今後の課題について述べる．

2. 関連研究

ネット上で負の影響をもたらす投稿の検出に関する研究は様々行われている．

石坂ら[2]は，電子掲示板サイトの投稿に対して，高いほどその単語が悪口単語であることを意味する単語悪口度を算出し，SVM を使用して閾値を超えるか否かで悪口文か否かを分類した．その結果，閾値が-0.2 のとき F 値が 0.90 で最大になり，悪口度が著しく低い単語を除くことで精度が向上することを報告した．

肥合ら[3]は，Twitter への日本語投稿を対象とし，皮肉に偏って出現する皮肉の特徴語を素性として皮肉ツイートの検出を行った．実験の結果，SVM を利用した二つの分類器の結果を比較し，よりよい出力結果を採用する分類手法と提案素性とを組み合わせることで F 値が最大で 0.85 を記録し，精度が向上することを報告した．

大友ら[4]は，いじめ表現辞書を作成することで，ネットいじめの自動検出を行った．いじめ表現辞書とは SO-PMI 値をいじめ度として付与した単語辞書であり，

いじめ度が高いほどその単語がいじめに関連する可能性が高いという意味をもつ．複数の機械学習手法に対し，いじめ表現を含む複数の特徴量を与えたところ，最大で検出精度の F 値 0.92 を記録した．また，ほかの特徴量に加えていじめ表現を使用することで精度が向上したことを報告した．

酒井ら[5]は，ディープニューラルネットワークを用いて怒り感情が含まれた日本語文書を検出する手法を提案した.酒井らはまず怒りの種類を 5 種類に分類し，さらにそれらを明示的怒りと暗示的怒りに分類した．そして,文書を構成する個々の文の怒り感情を CNN を用いた分類器により分類し，それらを統合して文書の怒り分類とした．その結果，2 段階検出を行った場合の精度が 0.38 で最大となった．

渡辺ら[6]は，Twitter のリプライに着目し，リプライのネガティブ度を求めることで炎上の種類を分類した．その結果，一定期間ごとのネガティブ度の変化を利用して炎上ツイートのクラスタリングを行う際に，期間が日ごとでは有効性を示す結果が得られなかったが，期間を時間ごとにすることで指定したクラスタ数に分類できたことを報告した．

本研究は，ニューラルネットワークを用いて明示的表現と暗示的表現が混在する煽りを含むツイートの検出を行うという点は酒井らの研究と類似している．しかし，分類対象とする文書全体を文単位に分割したうえで個々の文の属性を分類し，その結果を統合して改めて文書全体の属性の分類を行う酒井らの手法に対して，本研究では文書内の個々の文に着目することなく文書全文を一つの入力とみなしてそのまま分類を行うという点で異なる．

3. BERT による煽りツイートの分類

3.1. 煽りツイートの定義とそのラベル付け

本研究では煽り行為を「相手の感情を逆撫でる行為」と定義する．そして，煽りツイートを「内容が煽り行為のツイート」，すなわち「相手の感情を逆撫でる内容のツイート」と定義する．本研究で分類対象とするツイートは，リプライツイートのみもしくはリプライツイートとそのリプライ元のツイート組である．リプライを利用するのはリプライによる煽りの場合，煽り対象がリプライ元として存在し，またリプライ元が存在することでそのリプライに至る文脈が確認できるためである．ツイートの収集には TwitterAPI を利用する．その際，複数の単語のいずれかを指定し，収集する．よって収集されるのは指定した単語のいずれかを含むリプライツイート，およびそのリプライ元のツイートとの組となる．各ツイート組は煽りツイートか非煽りツイートか判定し，ラベル付けする．この判定は人手で行い,ラベルはリプライツイートが煽りか否かで「煽り」「非煽り」を付与する．

表 1：ツイート組とその煽り判定の例

リプライ元	リプライツイート	ラベル
おやすみでありますってリプしてませんでした？	うんで？そちらの意見はなんですか？	煽り
安倍政権の 2 倍以上の 20 年間委員長をしている志位氏はいつ終わるのでしょうか？	共産党政権は始まってすらいないのですが、僻み根性ですか？	煽り
親不知抜歯分の医療費、個別の会計はさほどだから気にしてなかったけど、年間にすると結構な額になっててびっくりした	ちりも積もれば…ですね。鎮痛剤とか…消毒通院とか手間もあるし…	非煽り

ラベル付けしたツイート組の例を表 1 に示す．表 1 の 1 つ目は直接的に煽る単語は含まれないが「うんで？」の表現からリプライ元に対して喧嘩腰であると判断し煽りとした．2 つ目は「僻み根性ですか？」が会話の流れからも煽りと読み取れるため煽りと判定した．3 つ目は煽るような単語もリプライ元との会話の流れから煽るような態度も見られなかったため非煽りとした．

3.2. Bidirectional Encoder Representations from Transformers (BERT)

3.2.1. BERT の概要

BERT は自然言語処理における事前学習モデルの一つである．大規模テキストコーパスから双方向 Transformer というニューラルネットワークの事前学習を行ったモデルを，個々のタスクに合わせてファインチューニングする．近年，広範囲の自然言語処理タスクにおいて最高水準の結果を示したことで，汎用的な言語モデルとして注目された．

図 1 に BERT の概略図を示す.$\boldsymbol{E_i}$が入力系列を表し，$\boldsymbol{T_i}$が出力系列を，Trm は Transformer[7]を表している．Transformer は Attention 機構を使用したニューラル機械翻訳モデルである．BERT では特に Self-Attention を利用して単語の重みを決定する．従来の言語表現モデルでは，OpenAI GPT[8]のように left-to-right モデルが用いられ，左から右に一方向のみで読み込み学習を行っていた.あるいは ELMo[9]のように left-to-right モデルと right-to-left モデルをそれぞれ独立に動かし,それぞれの出力を連結することで双方向の読み込みを実現していた．

一方 BERT は，同一のモデル内で双方向から単語の

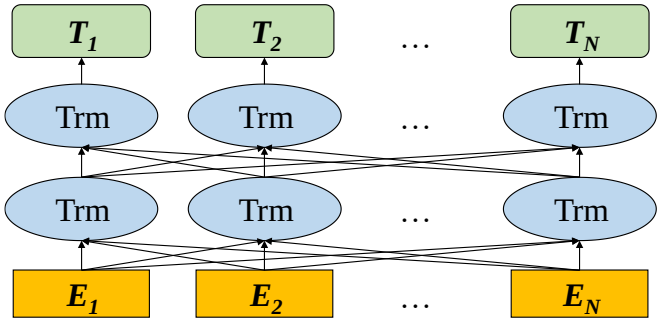

図 1：BERT の概略図 [1]

周囲の文脈を学習することを実現し，前後の関係性を考慮することができる．双方向での学習は，単方向での学習をそのまま双方向に適用すると学習時に予測すべき単語を先読みするため実現できなかったが，BERT では事前学習の際に，Masked Language Model (MLM)と Next Sentence Prediction (NSP)を用いて学習を行うことにより，予測する単語を先読みすることを防いでいる．

3.2.2. Masked Language Model (MLM)

MLM は入力シーケンスにある単語の 15%を[MASK]に置き換え，シーケンスにある単語のうちマスクされなかったものによって与えられる文脈に基づいて，[MASK]トークンに置き換えられた単語を予測するタスクである．例えば以下の文の場合，文中から無作為に選んだ単語"has"をマスクした文を作成する．その文に対して[MASK]トークンを推測する．

(1) My big sister has a bad mouth.
(2) My big sister [MASK] a bad mouth.

ただし，ファインチューニングには出てこない[MASK]トークンを事前学習では使用しているため，事前学習とファインチューニングの間に差異が生じる．そのため，事前学習ではマスクするトークンを常に[MASK]トークンに置き換えるのではなく，次のようにしてこの問題を緩和している．

- 80%の確率で，[MASK]トークンで置き換える
- 10%の確率で，ランダムに選んだトークンで置き換える
- 10%の確率で，そのままにして置き換えない

3.2.3. Next Sentence Prediction (NSP)

図 2 に NSP の概略図を示す．NSP は文のペアを受け取り，ペアにおいて 2 つ目の文が元の文章において後続の文になっているかを予測するタスクである．NSP の学習において BERT がペアとなっている 2 つの文を区別するために，ペアの文を入力として渡す前に図 2 のように埋め込み表現に置き換える．すなわち，図 2 中の Token Embeddings はトークンの ID を表し，Segment Embeddings は文のペアの区切れを表す．また Position Embeddings はシーケンス内の単語の位置を表している．それらを加算したベクトルのシーケンスが Transformer への入力となる．[CLS]は文の先頭を表すトークンであり，[SEP]は文の区切れを表す．また，BERT に入力される英文はサブワードに分割され，サブワードに分割された語の中で語幹でないものには“##”が付与される．例えば“playing”は“play”と“##ing”に分割される．事前学習において入力となる文のペアの 50%は，以下の(1)のようにオリジナルの文章中において自然に続く文であるが，残りの 50%のペアは後続文が以下の(2)のようにコーパスからランダムに選択された文である．

(1) [CLS] I’m looking [MASK] a subway station [SEP] the name is [MASK] station [SEP]
(2) [CLS] I’m looking [MASK] a subway station [SEP] they are out [MASK] stock [SEP]

3.3. BERT を用いた分類器の提案

BERT を利用した分類器は，データとしてリプライツイートのみを用いる分類器 S(Single)と，リプライ元と合わせたツイート組を用いる分類器 P(Pair)の 2 種類作成する．ツイート組を用いることで，リプライツイートの文面が同一でも文脈に応じて正しく分類することが期待される．

(S) リプライツイートの分類を行う分類器
(P) リプライツイートとそのリプライ元のツイートの組の分類を行う分類器

また，分類器を作成するためには BERT のモデルを用意する必要がある．しかし，煽り/非煽りの分類に特化したモデルを作成するための事前学習に十分な量のデータセットを用意するのは困難であるため，事前学習済みモデルをファインチューニングする．

3.4. ファインチューニングの方法

3.3 節で述べた通り，分類器は事前学習済みモデルをファインチューニングすることで作成する．BERT では，それぞれのタスクに応じた教師ありデータを用いてファインチューニングすることにより特定のタスクに特化したモデルを構成できるため，事前学習済みモデルをそのまま適用するよりも性能の向上が期待できる．事前学習済みモデルとしては BERT 日本語 Pretrained モデル[1]を使用する．また，入力テキストとなるツイートは Juman++[2]で形態素解析する．ファインチューニングに使用するデータは収集したツイート本

[1] http://nlp.ist.i.kyoto-u.ac.jp/index.php?ku_bert_japanese

[2] https://github.com/ku-nlp/jumanpp

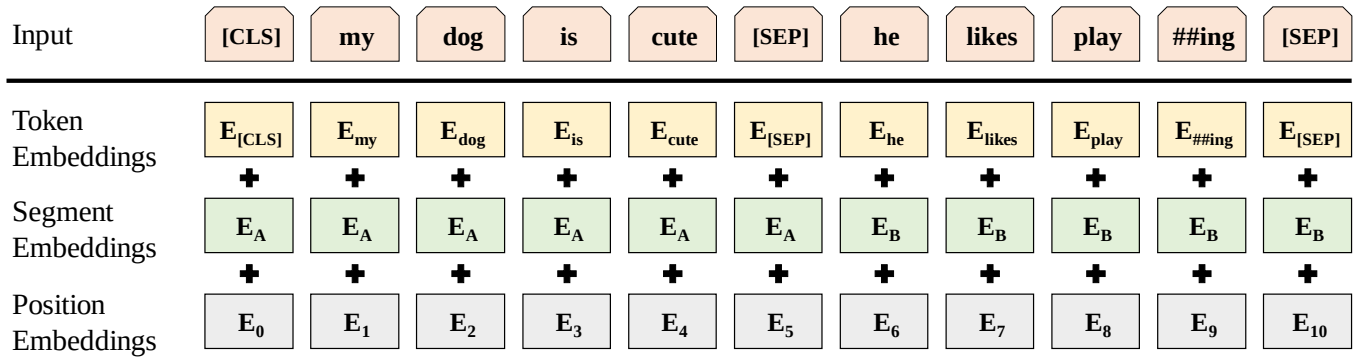

図 2：NSP の概略図 [1]

文である．分類器 S はリプライツイートのみを，分類器 P はツイート組をファインチューニングに用いる．

4. 煽りツイートの分類実験

4.1. 実験の概要

4.1.1. 実験に用いるデータ

実験には 3.1 節で説明したように TwitterAPI を利用して収集したツイート組を用いる．収集期間は 2020/09/06～2020/09/19 の 2 週間である．その際，表 2 に示す単語のいずれかが含まれているリプライツイートおよびそのリプライ元のツイートの組を収集した．

収集したツイート組に煽りまたは非煽りのラベル付けを行った結果，煽りツイート組 2134 件，非煽りツイート組 5652 件，計 7786 件のツイート組となった．

4.1.2. 実験内容

分類実験では，3.3 節で説明した分類器 S，P を用いて 4.1.1 項で述べたツイートまたはツイート組を分類し，BERT 以外の分類器と性能を比較する．データは分類器の説明通り，分類器 S はリプライツイートでファインチューニングを行い，リプライツイートのみを分類する．分類器 P はツイート組でファインチューニングを行い，ツイート組を分類する．これにより，リプライツイート単独とツイート組の場合で BERT の分類器の性能を比較する．

実施する実験は以下の三つである．

A) 煽りと非煽りのデータを同数とした二値分類
B) 煽りと非煽りのデータが同数ではない場合の二値分類
C) 煽りと非煽りそれぞれを細分類した場合の多値分類(分類器 S と P のみの比較)

実験 A で使用するデータは 4.1.1 項で述べたツイート組のうち，煽りツイート組は収集したすべての煽りツイートである 2134 組，非煽りツイートは収集した非煽りツイートから無作為に抽出した 2134 組，計 4268 組を用いる．実験 B，C では収集したツイート組の全てである 7786 組を用いる．

表 2：ツイート収集に使用した単語の一覧

です	ます	死ん
wwww	やめろ	だろ
でしょ	じゃね	しろ

実験 C では「煽り」を「強い煽り」と「弱い煽り」へ，「非煽り」を「煽りの要素を含む非煽り」と「煽りの要素を含まない非煽り」へさらに分け，「強い煽り」「弱い煽り」「非煽り」のみを利用した場合，「煽り」「煽りの要素を含む非煽り」「煽りの要素を含まない非煽り」のみを利用した場合，また細分類したすべてのラベルを利用した場合でそれぞれ分類実験を行う．

また，性能比較のために実験 A，B では scikit-learn[3] の以下の分類器を利用する．

- 非線形 SVM：非線形なデータを線形分離可能になる空間に写像し，超平面で線形分離することによりデータを識別する機械学習手法
- k-近傍法：識別したいデータ点から距離が近い k 点の学習データを探し，一番ラベルの多かったクラスを識別結果とする機械学習手法
- ランダムフォレスト：複数の決定木の多数決で識別結果を決める機械学習手法

いずれもリプライツイートのみで学習し，リプライツイートを分類する．

分類するツイートの日本語文を処理するため，入力テキストは Bag of Words (BoW)でベクトルに変換する．BoW は文書に単語が含まれているかのみを考えて，単語の順序は考慮しないモデルである．BoW に対して TF-IDF による重みづけを行ったのち，LSI(Latent Semantic Indexing) [10] により，300 次元まで次元圧縮する．文の単語単位での分かち書きには MeCab[4]を，べ

[3] https://scikit-learn.org/stable/

[4] https://taku910.github.io/mecab/

クトル変換，重みづけ，次元圧縮には gensim[5]を利用した．また，非線形 SVM，k-近傍法，ランダムフォレストの分類器のパラメータは下記の通り設定した．なお，記載のないパラメータはすべてデフォルトのままとした．

非線形 SVM 独自の設定として，正規化項の係数 C=10，rbf カーネルを使用し，カーネルの係数 gamma=0.1 である．k-近傍法独自の設定として，k=5 とし，近傍点の評価に距離も考慮する．ランダムフォレスト独自の設定として，決定木作成の評価指標はジニ係数，作成する決定木は 10，random_state=1，n_jobs=2 とした．また，クラスごとのサンプル数の偏りによる重みの違いは自動的に補正される．

分類器の性能は 5 分割交差検証で評価した．まずラベル付けしたツイート組を無作為に 5 分割し，うち 4 つを学習，残りの 1 つをテストに用いる．すなわち，全データの 8 割を学習，2 割をテストに用いる．また BERT を利用する分類器では，学習用データを 8 分割したうちの 1 つを検証に，残りをファインチューニングに用いる．すなわち全データの 7 割をファインチューニング，1 割を検証，2 割をテストに用いる．これを学習とテストに使う組を変えて 5 通りの組み合わせを作り，それぞれの場合の分類結果を確認する．その 5 回の平均を各分類器の評価に用いる．

各分類器の出力について，BERT の分類器による出力結果は「煽り」「非煽り」の各ラベルに対して 0 から 1 の範囲で正規化された確率であり，その総和は 1 になる．分類結果は入力に対する各ラベルの確率が最も大きいものをその出力とする．非線形 SVM は識別空間において「煽り」「非煽り」の各ラベルのどちらに対応する領域に属するかで分類する．k-近傍法は各近傍点のラベルと距離を合わせて評価し,「煽り」「非煽り」の二値で分類する．ランダムフォレストは各決定木で「煽り」「非煽り」の二値分類の結果を出力し，その多数決で勝利したラベルを最終的な出力とする．

4.1.3. 評価方法

分類器の評価には，5 分割交差検証による正解率(*Accuracy*)，再現率(*Recall*)，適合率(*Precision*)，F 値(*F-measure*) を利用する．それぞれ以下のように求める．

$$F\text{-}measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

なお，

TP:分類器が煽りと判断した煽りツイート

FP:分類器が煽りと判断した非煽りツイート

FN:分類器が非煽りと判断した煽りツイート

TN:分類器が非煽りと判断した非煽りツイート

である．

F 値は再現率と適合率の調和平均で，再現率はすべての煽りツイートに対し，分類器が煽りであると判断したものの割合，適合率は分類器が煽りであると判断したデータのうち，実際に煽りであるものの割合である．煽りツイートの検出という観点では，再現率はどれだけ煽りを見逃さないか，適合率はどれだけ誤検出を防げるかをそれぞれ示しており，F 値はそれらを総合的に評価した指標といえる．

ラベルの種類を増やして細分類したツイートを分類する実験 C では，最多の場合「強い煽り」「弱い煽り」「煽りの要素を含む非煽り」「煽りの要素を含まない非煽り」の 4 種類のラベルが現れる．しかし評価はラベルが「煽り」「非煽り」のどちらに属するかで行う．例えば，正解が「強い煽り」のものに対して分類器の出力が「弱い煽り」であった場合，ラベルは一致していないが正解と出力がともに「煽り」に属するため上述の *TP* とみなす．

4.2. 実験結果

4.2.1. 実験 A：煽りと非煽りデータ数が同数

作成した分類器の分類結果を表 3 に示す．表の太字は各評価指標について最も大きい値を示している．

表 3 より，正解率と適合率，F 値に関しては BERT を利用した分類器 S が最も高かった．再現率は非線形 SVM が最も高かった．

BERT を利用したものと利用していないものの各評価指標について比較する．まず，k-近傍法，ランダムフォレストに関してはどの評価指標に関しても他の分類器に勝っている部分がない．よって残りの分類器 S と P，非線形 SVM を比較する．正解率に関しては分類器 S と P はいずれも約 0.72 だが非線形 SVM は 0.678 とやや劣る．再現率は非線形 SVM が 0.721 と最も高い値を示しているが分類器 S の 0.706 や分類器 P の 0.700 との間に大きな差はない．適合率は正解率と同様に分類器 S と P ともに約 0.72 で，非線形 SVM が 0.666 とやや劣る．F 値は分類器 S が 0.715，つづいて分類器 P の 0.711，少し劣って非線形 SVM の 0.693 である．これらの結果から，リプライツイートのみを使用した分

[5] https://radimrehurek.com/gensim/

類器 S はツイート組を使用した分類器 P をわずかに上回ったが，いずれの評価指標においてもその差は 0.01 未満とほとんどないことがわかる．

4.2.2. 実験 B：煽りと非煽りデータ数が同数でない

作成した分類器の分類結果を表 4 に示す．表 4 より，正解率に関しては BERT を利用した分類器 S が，再現率は分類器 P が最も高かった．適合率は非線形 SVM が最も高く，F 値は分類器 S と P が最も高かった．

正解率に関しては分類器 S と P はともに約 0.77 で，つづいて非線形 SVM は 0.750 とやや劣る．k-近傍法とランダムフォレストはともに約 0.71 とさらに劣る．再現率は分類器 P が 0.552 と最も高いが，分類器 S の 0.549 との間に差はほとんどない．それに k-近傍法が 0.329，ランダムフォレストが 0.313 とつづき，非線形 SVM が約 0.184 と最も低くなった．適合率は非線形 SVM が 0.683 と最も高い値を示し，つづく分類器 S が 0.596，分類器 P が 0.594 となっている．k-近傍法とランダムフォレストはともに約 0.47 とさらに劣る．F 値は分類器 S，P ともに 0.571 で，残りの分類器は大きく劣る．これらの結果から，リプライツイートのみを使用した分類器 S とツイート組を使用した分類器 P の間に差がほとんどないことがわかる．

4.2.3. 実験 C：煽りと非煽りの細分類の利用

作成した分類器の分類結果を表 5 に示す．出力結果は 4.1.3 項で述べたように，細分類したラベルが煽りのグループか非煽りのグループかのどちらかで判定する．表の太字は各評価指標に関してそれぞれの分類器で最も大きい値を，下線は各評価指標に関して全体で最も大きい値を示している．

表 5 に示す通り，煽りと非煽りともに 2 クラスに細分類した場合，正解率と再現率が向上しているが，適合率と F 値は大きく減少している．非煽りのみ細分類した場合，どちらの分類器も F 値が最大になっている．一方で煽りのみを細分類した場合，細分類しない場合と比べて分類器 S の F 値はわずかに高かったが，分類器 P の F 値はわずかに低かった．

5. 考察

5.1. 各分類器の結果

まず，表 3 と表 4 より，k-近傍法とランダムフォレストに関してはいずれの指標に関しても分類器 S も P も上回っていないため，いずれかの指標で上回っているものがあった非線形 SVM と比較する．

表 3 より煽りと非煽りが同数の場合，非線形 SVM が唯一再現率で BERT を利用した分類器 S と P を上回る結果を示している．しかし，分類器 S との差は 0.015，一方適合率では 0.058，F 値では 0.022 の差をつけられていることを考慮すると，非線形 SVM が BERT を利用した分類器 S･P より高い性能を持つとはいえない．

表 3：実験 A の分類結果

	正解率	再現率	適合率	F 値
BERT (S)	**0.719**	0.706	**0.724**	**0.715**
BERT (P)	0.716	0.700	0.723	0.711
非線形 SVM	0.678	**0.721**	0.666	0.693
k-近傍法	0.600	0.644	0.595	0.615
ランダム フォレスト	0.617	0.595	0.623	0.610

表 4：実験 B の分類結果

	正解率	再現率	適合率	F 値
BERT (S)	**0.774**	0.549	0.596	**0.571**
BERT (P)	0.773	**0.552**	0.594	**0.571**
非線形 SVM	0.750	0.184	**0.683**	0.289
k-近傍法	0.711	0.329	0.466	0.385
ランダム フォレスト	0.714	0.313	0.472	0.376

表 5：実験 C の分類結果

		正解率	再現率	適合率	F 値
煽り 1 クラス 非煽り 1 クラス	S	0.774	0.549	0.596	0.571
	P	0.773	0.552	0.594	0.571
煽り 1 クラス 非煽り 2 クラス	S	0.777	0.570	0.600	**0.584**
	P	0.777	0.569	**0.600**	**0.583**
煽り 2 クラス 非煽り 1 クラス	S	0.779	0.545	**0.607**	0.574
	P	0.767	0.550	0.580	0.564
煽り 2 クラス 非煽り 2 クラス	S	**0.787**	**0.830**	0.202	0.324
	P	**0.793**	**0.854**	0.209	0.335

表 4 より煽りと非煽りが同数ではない場合，非線形 SVM が適合率では分類器 S を 0.087 上回っている．そのため非線形 SVM は煽りの誤検出の防止に関しては有効な可能性がある．しかし，再現率が 0.184 と非常に小さいため煽りツイートの検出に有用とはいえない．

ここで分析のために，4.2.2 項の実験 B のテストデータの一つのリプライツイートの主成分分析を行った．その結果をプロットし，点が密集していた縦軸・横軸ともに[-2.0, 2.0]の範囲を図 3 に示す．

主成分分析では，非線形 SVM 等への入力データである 300 次元のベクトルを主成分分析した．図 3 の PC1 は第一主成分，PC2 は第二主成分，軸の数値は標準化されたベクトルの値である．黄色の点は正解の煽りツイート，紫の点は非煽りツイートを示している．

非線形 SVM は非線形データを異なる空間に写像し，超平面により線形分離する機械学習手法であるが，近い距離に異なるラベルのデータ点が密集していると精度よく分離できない．

k-近傍法は識別したいデータ点から距離が近い k 点の学習データを探し，各ラベルに対してその個数を比較する手法であるが，非線形 SVM 同様に異なるラベ

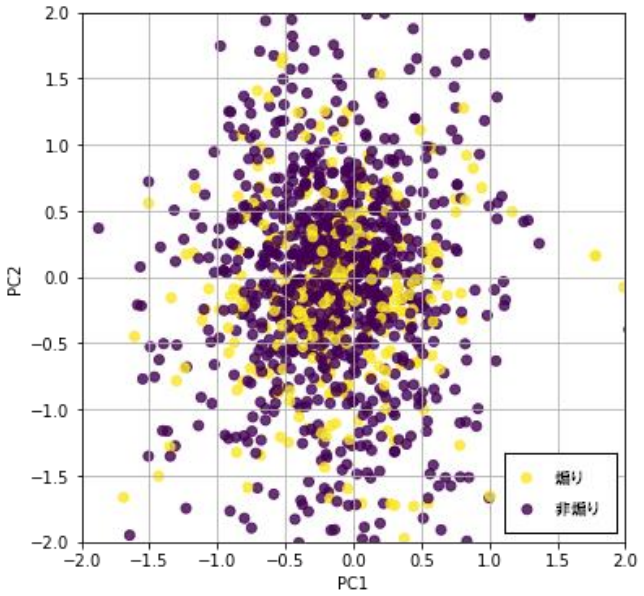

図 3：テストデータ(実験 B)の主成分分析の結果

ルのものが密集していると精度の高い識別ができない．したがって，図 3 のデータ分布において，両ラベルが混在し密度の高い位置に属するデータ点に関しては正しく識別することは困難である．

ランダムフォレストは単純な識別条件で分類を繰り返し，最終的にラベルを決定する手法であるが，作成する決定木の数や深さが十分でない場合，精度は向上しない．

5.2. リプライツイートのみとツイート組の分類性能

4.2 節の実験結果から，リプライツイートのみを分類する分類器 S は二つのラベルのデータが同数でも同数でなくても，リプライ元と合わせた組の分類である分類器 P と比べて性能に大きな差がみられなかった．そこでここでは，リプライツイートのみの分類とリプライ元と合わせたツイート組の分類の違いについて考察する．

表 6 に実験 B のテストデータの全てに対する分類器 S，P の平均正解数を示す．表の各行は正解ラベルが煽りと非煽りのそれぞれのデータに関して，各分類器が正解したか否かを示しており，数値は 5 分割交差検証で行った 5 回の分類の平均正解数である．S のみ正解，P のみ正解がともに存在していることから単純にいずれかがよいというわけではないことがわかる．次に，分類されたツイートの違いについて着目する．

表 7 にテストデータ中のリプライの文面が「wwww」という同一のツイート組を示す．分類器 S はテストデータ中のリプライが同一文面のデータには同一の判定をする．一方，分類器 P はリプライ元と合わせて評価し，リプライが同一文面のツイート組に対して異なる

表 6：分類器 S と P の平均正解数

	ともに正解	S のみ正解	P のみ正解	ともに不正解
煽り	194.0	40.4	41.6	150.8
非煽り	912.0	59.2	56.0	103.2

表 7：リプライが同一文面のツイート組の例

リプライ元	リプライ	ラベル	判定
もっと褒めろ！！	wwww	煽り	ともに正解
あまりに記憶が曖昧で え？ 本人？って疑われまくった 30 分間おもろ	wwww	非煽り	P のみ正解
センス〇いな	wwww	煽り	S のみ正解

判定をし，正しく分類できたものがある．また，5 回の試行の中でリプライが「wwww」という同一文面のデータに対して，どちらか一方の分類器のみが正しく判定したものを数えたところ，分類器 S のみ正解が 7 件，分類器 P のみ正解が 12 件であった．よってリプライ元と合わせたツイート組を利用することで，リプライの文面のみからでは分からない煽りの分類に一部成功していることがわかる．

また，どちらの分類器でも正しく分類できなかった煽りツイートも 3 割近く存在した．そのため，本手法で用いたツイート本文の特徴だけでは捉えられなかった煽りもまだ多くあるといえる．さらなる検出のためにはユーザ同士のフォロー・被フォロー関係のような他の特徴も取り入れる必要がある．

6. まとめ

本研究では，煽りツイートの検出のため，BERT を利用してリプライツイートおよびそのリプライ元との組を煽りか非煽りかに分類する分類器を作成し，それらの分類性能を実験により評価した．

具体的には，リプライツイートとツイート組を使用して日本語事前学習済みの BERT のファインチューニングを行い，分類器の評価実験を行った．評価実験の結果，煽りと非煽りのデータ数が同数の場合，BERT を利用してリプライツイートのみで学習し，分類した結果が F 値 0.715 を示し最も高かった．しかし，ツイート組の分類をした結果の F 値が 0.711 と差はほとんどなかった．また煽りと非煽りのデータ数が同数でない場合，BERT を利用してリプライツイートの分類をした結果もツイート組の分類をした結果もともに F 値 0.571 を示し最も高く，両者の間に差はなかった．これらの結果より，従来の機械学習手法と比較して BERT を利用した分類器は煽りツイート検出に有効であった．

また，ツイート組でファインチューニングした分類器はリプライツイート単独のものと比較して，リプライツイートが同一文面でラベルが異なるツイート組の一部を正しく分類した．

今後の課題としては検出精度の向上のために，ほかの特徴を取り入れることが挙げられる．例えばリプライを行った人物とリプライ元の人物の関係性や，ツイートの投稿時刻などを特徴として追加することを検討したい．

参 考 文 献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[2] 石坂達也，山本和英．Web 上の誹謗中傷を表す文の自動検出．言語処理学会 第 17 回年次大会，E1-6，2011.

[3] 肥合智史，嶋田和孝．偏りのある特徴語を考慮した皮肉の検出．言語処理学会 第 23 回年次大会，P17-3，2017.

[4] 大友泰賀，張建偉，中島伸介，李琳．いじめ表現辞書を用いた Twitter 上のネットいじめの自動検出．DEIM2020，C7-1，2020.

[5] 酒井優介，藤ノ木太郎，安藤雅洋，湯川高志．ディープラーニングを用いた暗示的怒りの自動検出手法．第 33 回人工知能学会全国大会，4M3-J-9-04，2019.

[6] 渡辺みずほ，佐藤哲司，”リプライのポジネガ極性を用いた Twitter 炎上の分類手法の提案”，DEIM2020，C7-3，2020.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems 30, pp. 5998–6008, 2017.

[8] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. Technical report, OpenAI. 2018.

[9] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1810.04805, 2018.

[10] Deerwester, S., Dumais S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, Vol. 41 ,pp. 391-407, 1990.