ファインチューニングを用いた畳み込みニューラルネットワークによる皮肉検出

Ohara, Kotaro / 大原, 虎太郎

(出版者 / Publisher)
法政大学大学院理工学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 理工学研究科編
(巻 / Volume)
64
(開始ページ / Start Page)
1
(終了ページ / End Page)
8
(発行年 / Year)
2023-03-24
(URL)
https://doi.org/10.15002/00026390

法政大学大学院紀要　理工学研究科編　Vol.64(2023 年 3 月）　　法政大学

ファインチューニングを用いた畳み込みニューラルネットワークによる皮肉検出

SARCASM DETECTION USING CONVOLUTIONAL NEURAL NETWORK WITH FINE-TUNING

大原虎太郎
Kotaro Ohara
指導教員　平原誠

法政大学大学院理工学研究科応用情報工学専攻修士課程

Sarcasm detection is an important task in correctly understanding a sentence. This study proposes a method of sarcasm detection using a Convolutional Neural Network (CNN) with fine-tuning. Assuming sarcasm is caused by sentiment, we attempt to pre-train the CNN on a sentiment analysis task to determine whether a sentence is negative or positive. We fine-tune the pre-trained CNN on the sarcasm detection task to determine whether a sentence is sarcastic or non-sarcastic. We found that the proposed method could detect sarcasm more accurately than CNN, Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) with randomly initialized parameters.

Key Words *: Sarcasm Detection, Sentiment Analysis, Neural Networks, Fine-tuning*

１．はじめに

近年，人間と機械のコミュニケーションに関する研究が数多く行われており，音声による命令を実行する音声アシスタントなども広く普及し始めた．これらは非常に便利で画期的ではあるが，「会話」という観点からみるとそのレベルはまだ低く，人間同士のような自然な会話を実現するには至っていない．

人間と機械の自然な会話における大きな壁の１つに，「皮肉」がある．皮肉は「相手の欠点・弱点を遠回しに非難したり，からかったりすること」と定義されており，文章を言葉通りに捉えるだけでは，その意味を正しく理解することは難しい．例えば，

「これほど掃除しがいのある家は初めてだ.」

という文章において，話し手は掃除のやりがいがあるという肯定的な表現を用いて，家がひどく乱雑であるということを表している．また，

「彼は生徒に大量の課題を出す素晴らしい先生だ.」

という文章においては，話し手は「素晴らしい」という肯定的な表現を用いて非難している．このように，皮肉はその文章の意味を反転させてしまう可能性があるため，意味の正しい理解において皮肉検出は重要なタスクである．

皮肉検出を目的とする先行研究は数多く行われている[1], [2], [3]．日本語の皮肉検出の先行研究では，その多くが統計ベースやルールベースで行われている．例えば先行研究[4]では，皮肉の文章で肯定表現が否定的な意味を持つようになる使われ方を分析することで 8 つの皮肉クラスを定義し，それぞれのクラスについて作成した極性や構文パターン，頻出フレーズ，婉曲表現，強調表現に基づく皮肉検出ルールと例外ルールを文章に対して段階的に適用することで皮肉の検出を行う手法が提案されている．機械学習ベースで行われている先行研究では，回帰結合によって時系列を扱うことができる再帰型ニューラルネットワーク（RNN : Recurrent Neural Network）が使われている．例えば先行研究[5]では，批判する側と批判される側の立場をベクトルで表現し，RNN ベースのネットワークに追加入力することで皮肉の検出を行う手法が提案されている．また，近年は BERT を用いた手法を提案する先行研究もあり，先行研究[6]では F 値 0.82 の識別精度を出している．一方で，一般的に画像処理に用いられる畳み込みニューラルネットワーク（CNN : Convolutional Neural Network）を用いた先行研究[7]があり，英語の皮肉検出において高い精度を出している．

本研究では，「皮肉は感情に起因する」と仮定し，まず CNN による感情分析の事前学習を行う．それによって得られる感情の特徴を捉えた学習済みパラメーターを，目的タスクの皮肉検出 CNN でファインチューニングすることで皮肉の検出を行う．

２．単語分散表現

本節では，テキストを CNN で処理するために用いる単

語分散表現の獲得について述べる．

２.１．単語分散表現

画像は RGB などの画素値，音声などの音は音圧のように，数値として扱うことができる．一方，言語は人間が定義した記号であるため，そのまま数値化することができない．そのため言語を機械的に処理するためには，まず数値化する必要がある．

単語を数値化する方法の 1 つに「one-hot ベクトル表現」がある．これは，全ての要素のうち 1 つの要素だけが 1 になり，その他の要素は全て 0 であるベクトルで表現する手法である．こうすることで，単語とベクトルが一対一対応することになり，機械的に処理することが可能になる．しかし，ボキャブラリが増加するほどベクトルの次元が膨大になり計算時間が大幅に増加する．また，単語同士の関係を表現できない，単語の追加が難しいなどの問題点がある．

one-hot ベクトル表現の問題点を解決したのが，「単語分散表現」である．単語分散表現では，単語は同じ空間の点として捉えられるため，単語同士の関係を表現することができる．すなわちベクトル同士の演算によって関係を表現できるという特徴を持つ．加えて one-hot ベクトルで表現する手法と比べ低次元で多くの単語を表現することが可能である．

単語分散表現を作成することのできるニューラルネットワークモデルに Word2Vec がある[8]．モデルは，入力層，中間層，出力層の 3 層からなる．Word2Vec には skip-gram と CBoW の 2 種類のアーキテクチャがあり，skip-gram はある文章において注目する単語（注目単語）からその前後の単語を予測するように学習させるのに対し，CBoW は注目単語の前後の単語からその注目単語を予測するように学習させる．そしてこの時のモデルの重みが単語分散表現となる．

本研究では，テキストを数値化して機械的に処理できるようにするため，Word2Vec により単語分散表現を作成する．Word2Vec のアーキテクチャは CBoW を選択した．これは，CBoW が ski-gram と比較して学習が高速であるからである．

２.２．単語分散表現作成のデータセット

モデルを学習するためのテキストには，Wikipedia の日本語記事全文を使用した．本研究で使用した記事は 2021 年 6 月 13 日時点で最新のものである．

２.３．テキストの前処理

テキストに対して，「クリーニング」，「正規化」，「分かち書き」の 3 種類の前処理を行った．

①　クリーニング

クリーニングでは，絵文字や顔文字といった記号の除去を行った．これらは感情を表す記号であるため感情分析の大きな手がかりとなるが，その数は膨大であり，新しいものも次々と作られるため，分散表現の獲得が難しい．そのため，テキストから除去した．

表 1　単語分散表現の評価

	本研究の単語分散表現	日本語 Wikipedia エンティティベクトル
評価	0.512	0.499

②　正規化

正規化では，テキストの表記揺れを無くすための文字列変換を行った．例えば，全角「リンゴ」と半角「ﾘﾝｺﾞ」は同じ意味を持つが，そのまま単語分散表現を作成すると分散表現はそれぞれ異なるものになる．このような問題は表記揺れと呼ばれ，半角全角の統一や，アルファベットの大文字小文字の統一，組文字（㌔や㍿）の変換を行うことで同じ単語として処理されるようにする必要がある．また，数字は組み合わせによって無限のバリエーションを持つため，全ての数字列の分散表現を獲得することは難しい．そのため数字は全て 0 に変換した．

③　分かち書き

「分かち書き」は，テキストを意味を持つ最小単位である形態素に分割する処理である．英語は，単語と単語の間が空白で区切られており，単語の同定が容易である．一方日本語は，単語と単語の間が空白で区切られていないため，そのまま機械的に処理しようとすると単語の同定が困難である．そのため，日本語の自然言語処理では，単語と単語の間に空白を入れる分かち書きを行う．例えば，

「私はリンゴが好きです」

というテキストに対して分かち書きを行うと

「私　は　りんご　が　好き　です」

となる．本研究では，形態素解析エンジン「MeCab」[9]を用いて分かち書きを行った．

２.４．Word2Vec の学習

前処理したテキストをモデルに入力し，200 次元の単語分散表現を獲得した．モデルの作成には，自然言語処理の分野で広く利用されているライブラリ「Gensim」[10]を使用した．

２.５．単語分散表現の評価

単語分散表現の性能を評価する方法は主に 2 種類ある．1 つは，単語同士の類似度や関連度を測る方法である．これを行うためのデータセットの 1 つに，「日本語単語類似度・関連度データセット JWSAN」[11]がある．評価用データセットの単語ペアの類似度と，当該ペアの単語分散表現間の類似度から，スピアマンの順位相関係数を算出することで評価スコアとする．もう 1 つは，アナロジータスクを利用する方法である．アナロジータスクとは，単語分散表現では単語同士の演算が可能という特徴を利用し，

$$king - man + woman = queen$$

というように，単語を類推するタスクである．

本研究では，1 つ目の単語の類似度，関連度を測る方法を用いて単語分散表現の性能評価を行った．評価用デー

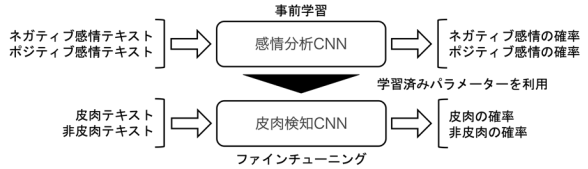

図 1　提案手法の概略

タセットには，JWSAN の JWSAN-1400 を用いた．性能評価の結果を表 1 に示す．結果から，同じ 200 次元の単語分散表現であり公開されている「日本語 Wikipedia エンティティベクトル」[12]と同程度の評価であることがわかる．

３．提案手法

本研究の提案手法の概略を図 1 に示す．本研究では，CNN を使用し，あらかじめ感情の特徴を捉えたパラメーターを再学習させることによって皮肉検出を行う．「皮肉は感情に起因する」と仮定し，テキストのネガティブ感情とポジティブ感情を識別する CNN（以下，感情分析 CNN）の事前学習を行い，それを用いてテキストの皮肉，非皮肉を識別する CNN（以下，皮肉検出 CNN）をファインチューニングして皮肉の検知を行う．

ファインチューニングとは，機械学習における学習手法の 1 つである．教師あり学習は，トレーニングデータと教師データのペアを与えることで学習を行うが，タスクによっては大量のデータと膨大な学習時間が必要となる．ファインチューニングは，目的タスクに関連する別タスクの学習によって得られた知識を目的タスクに転用するという考えに基づいており，別タスクの学習済みモデルのパラメーターの一部もしくは全てを，目的タスクの未学習モデルのパラメーターの初期値とし，再学習，微調整する手法である．十分なデータと学習時間で別タスクの事前学習を行い，目的タスクにおいてファインチューニングすることで，少ないデータや学習時間でも高い精度を出すことが可能であり，その有効性は多くの自然言語処理のタスクにおいて示されている[13], [14].

３.１．感情分析

本項では事前学習である感情分析において使用するデータセットとネットワークについて説明する．

３.１.１．感情分析データセット

事前学習の感情分析では，「Twitter 日本語評判分析データセット」[15]を使用した．これは，Twitter に投稿された携帯電話や家電に関するテキストのデータセットであり，「Xperia」や「AQUOS」，「iPhone」といった 8 種類のジャンルのテキストを提供している．それぞれのテキストには，該当するジャンルに対する「ネガティブ」および「ポジティブ」のラベルに加え，ネガティブな内容とポジティブな内容が共存する場合には「ネガティブ&ポジティブ」，ネガティブな内容もポジティブな内容も書かれていない場合には「ニュートラル」，ジャンルに無関係な内容が書かれている場合には「無関係」の 5 種類のラベル

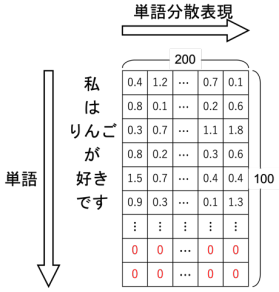

図 2 テキストの数値化

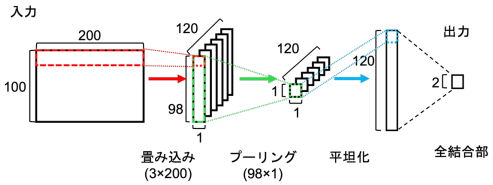

図 3 感情分析 CNN

が付与されている．本研究では，この中からデータ数とラベルのバランスが最も良いジャンル「Xperia」を選択し，「ネガティブ」と「ポジティブ」のラベルが付与されているテキストを使用した．テキストの数はネガティブ 5,770 件，ポジティブ 3,499 件であり，その中からそれぞれ 3,200 件ずつを使用した．

テキストには，前述の前処理①～③に加え，単語分散表現を用いてテキストを数値化する処理④を行い入力データを作成した．数値化の例として，分かち書きしたテキスト

「私　は　りんご　が　好き　です」

に対する単語分散表現への変換を図 2 に示す．変換方法は先行研究[7]に従い，テキストを分かち書きした単語ごとに縦方向に並べ，各単語の分散表現を横方向に並べることで行った．この場合，テキストごとに単語数が異なるため，入力データの入力長（縦幅）が変わってしまう．この問題には，あらかじめ入力長を 100 単語と定め，空白となる部分は 0 の値で埋めることで対処した．本研究で作成した単語分散表現は 200 次元であるため，入力データは縦 100 次元，横 200 次元となる．なお，空白を 0 で埋める処理をテキストの前後で 2 通り行うことで，入力データをそれぞれ 2 倍の 6,400 件に水増しした．実験では，トレーニングデータを 5,120 件，バリデーションデータとテストデータをそれぞれ 640 件に分割して使用した．

３.１.２．感情分析 CNN

感情分析 CNN のネットワーク構造を図 3 に示す．ネットワーク構造およびハイパーパラメーターの最適化は，ハイパーパラメーター自動最適化フレームワークの Optuna[16]を用いて行った．ネットワークは，1 層の畳み込み層，1 層のプーリング層，2 層の全結合層からなる．データが入力されると，サイズ3×200のフィルターを 120 枚使用し，ストライド 1 で畳み込みが行われる．これにより，単語の前後関係の特徴が抽出される．畳み込み層

表 2　皮肉検出データセット 2 において除去したテキストとその理由の例

ラベル	テキスト	除去理由
皮肉	粘ればいつか勝利できると教えてくれたお三方ありがとう！勉強になりました	ラベルの正誤が不明
皮肉	VﾊﾞﾙﾃｨﾉBﾎﾞｯｸｽ グランプリすごすぎるな レッドブルリンク時のＭＡＣＬＡＲＥＮの応援ありがとうと似ている	データとして適していない
非皮肉	あびゃびゃびゃびゃびゃびゃびゃ	データとして適していない

の活性化関数には ReLU 関数を使用した．畳み込み層からの出力はプーリング層に入力され, サイズ98 × 1のウィンドウを使用し, 最大値プーリングによって特徴集約, 次元圧縮が行われる．プーリング層からの出力は平坦化され，全結合層に入力される．全結合層の 1 層目は 120 次元である．2 層目の出力層では softmax 関数を使用し，ネガティブである確率とポジティブである確率の 2 つが出力される. 最適化には Adam を使用し, 誤差関数には交差エントロピー誤差を使用した.

３.２．皮肉検出

本項では皮肉検出において使用するデータセットとネットワークについて説明する.

３.２.１．皮肉検出データセット

皮肉検出では，公開されている日本語の皮肉検出タスク用のテキストデータセットが存在しないため，TwitterAPI[17]を利用して Twitter の投稿から皮肉テキストを収集し使用した．Twitter を使用した英語の皮肉検出の研究の多くは，「#sarcasm」というハッシュタグをキーワードにテキストを収集している. これにより, 多くの皮肉テキストを収集することが可能である. しかし, 日本語の投稿において「#皮肉」というハッシュタグが付けられた投稿は極めて少ないため，先行研究[5]に従い，「（皮肉）」という言葉をキーワードとして収集した. 非皮肉テキストとしては，「（皮肉）」という言葉が含まれない投稿をランダムに収集した.

本研究は独立した文章から皮肉の検出を行うことを目的としており，会話の流れを考慮した皮肉検出は研究の範囲外である. そのため, 投稿へのリプライは会話の流れを前提とした内容である可能性が高いため，収集の対象外とした．また収集する投稿の対象はテキストのみの投稿とし，URL や画像，動画が含まれる投稿も収集の対象外とした．これは，URL であればリンク先の内容，画像であれば画像の内容が，テキストの内容の前提になってしまう可能性が高いからである.

これらの方法によって収集したテキストから, 2 種類の皮肉検出データセットを作成した.

（1）皮肉検出データセット 1

皮肉検出データセット 1 は，収集したテキストの各々を著者が確認し，アノテーションの訂正やノイズとなりうるテキストの除去を行ったデータセットである．Twitter には「皮肉をツイートする際は必ず（皮肉）という表現を使用する」という明確なルールがあるわけではない. そのため前述の収集方法だと, 皮肉テキストが非皮肉テキストとして収集される場合があり，その逆の場合もある．また，Twitter は日本語表現の自由度が非常に高く, 誤った文法が使われているテキストや, そもそも日本語として成立しておらず学習データとして適さないテキストが多く存在する. そのため, 収集したテキストに対して, 皮肉, 非皮肉のラベルが適しているかの確認と訂正を行い, また誤った日本語が使用されているテキスト, 学習データとして適さないテキストの除去を行い，実験に理想的なデータセットを作成した．除去したテキストとその理由の例を表 2 に示す.

テキスト除去後のテキスト数は皮肉, 非皮肉ともに 640 件である. これに対して, 感情分析データセットと同様の前処理①～④を行い，それぞれ 1,280 件ずつの入力データを作成した. 実験では, トレーニングデータを 1,024 件, バリデーションデータとテストデータをそれぞれ 128 件に分割して使用した.

（2）皮肉検出データセット 2

皮肉検出データセット 2 は，皮肉検出データセット 1 の作成において行ったノイズとなりうるテキストの除去を行わず，収集したテキストをそのまま使用した実環境に近いデータセットである. 皮肉検出データセット 2 は，実環境に近い環境で提案手法がどの程度堅実に精度を出すことができるかを確かめるために用いる.

テキスト数は皮肉, 非皮肉ともに 2,900 件である．これに対して，感情分析データセットと同様の前処理①～④を行い，それぞれ 5,800 件ずつの入力データを作成した. 実験では，トレーニングデータを 4,640 件，バリデーションデータとテストデータをそれぞれ 580 件に分割して使用した.

３.２.２．皮肉検出 CNN

皮肉検出 CNN は, Optuna を用いてデータセットごとにネットワーク構造およびハイパーパラメーターの最適化を行った.

（1）皮肉検出データセット 1 の皮肉検出 CNN

学習済みネットワークのパラメーターは，出力層に近いものほどタスク固有になる場合が多い. そのため, 畳み込み層のパラメーターのみ感情分析 CNN の学習済みパラメーターで初期化し，全結合層のパラメーターはランダムな数値で初期化した．皮肉検出データセット 1 を用

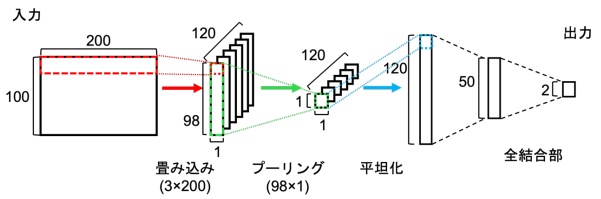

図 4　皮肉検出データセット 1 の皮肉検出 CNN

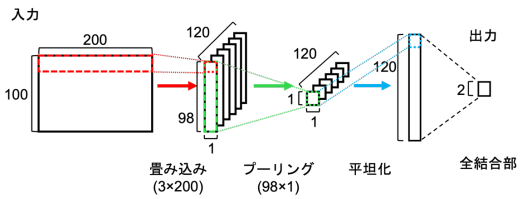

図 5　皮肉検出データセット 2 の皮肉検出 CNN

表 3　感情分析の評価

	正解率	適合率	再現率	F 値
感情分析 CNN	0.851	0.872	0.822	0.846

いた実験で使用した皮肉検出 CNN のネットワーク構造を図 4 に示す．ネットワークは，1 層の畳み込み層，1 層のプーリング層，3 層の全結合層からなる．データが入力されると，サイズ3×200のフィルターを 120 枚使用し，ストライド 1 で畳み込みが行われる．これにより，単語の位置関係の特徴が抽出される．畳み込み層の活性化関数には ReLU 関数を使用した．畳み込み層からの出力はプーリング層に入力され，サイズ98×1のウィンドウを使用し，最大値プーリングによって特徴集約，次元圧縮が行われる．プーリング層からの出力は平坦化され，全結合層に入力される．全結合層のノード数は 1 層目が 120，2 層目が 50，3 層目が 2 である．3 層目の出力層では softmax 関数を使用し，皮肉である確率と非皮肉である確率の 2 つが出力される．

（2）皮肉検出データセット 2 の皮肉検出 CNN

データセット 1 の CNN 同様，畳み込み層のパラメーターのみ感情分析 CNN の学習済みパラメーターで初期化し，全結合層のパラメーターはランダムな数値で初期化した．データセット 2 を用いた実験で使用した皮肉検出 CNN のネットワーク構造を図 5 に示す．ネットワークは，1 層の畳み込み層，1 層のプーリング層，2 層の全結合層からなる．データが入力されると，サイズ 3× 200のフィルターを 120 枚使用し，ストライド 1 で畳み込みが行われる．その後の処理は(1)と同様である．全結合層のノード数は 1 層目が 120，2 層目が 2 である．2 層目の出力層では softmax 関数を使用し，皮肉である確率と非皮肉である確率の 2 つが出力される．

4．実験

本節では，感情分析と皮肉検出の結果を示す．

4.1．感情分析

感情分析データセットのトレーニングデータを感情分析 CNN に入力し学習を行った．感情分析 CNN のテストデータに対する正解率，適合率，再現率，F 値を表 3 に示す．学習は 10 回行い，数値はその平均値である．この感情分析 CNN の学習済みパラメーターを使用し，皮肉検出 CNN のファインチューニングを行う．

4.2．皮肉検出

本項では，まず比較実験用ネットワークについて説明し，その後提案手法と比較実験用ネットワークの実験結果について述べる．

4.2.1．CNN

皮肉検出タスクにおける感情の特徴を捉えたパラメータをファインチューニングすることの有効性を確認するため，ファインチューニングを行わず，全てのパラメーターをランダムな数値で初期化し学習を行う CNN を作成した．ネットワーク構造およびハイパーパラメーターは，皮肉検出 CNN と同様に Optuna を用いてデータセットごとに最適化した．

（1）皮肉検出データセット 1 の CNN

皮肉検出データセット 1 を用いた実験で使用した CNN は，1 層の畳み込み層，1 層のプーリング層，4 層の全結合層からなる．サイズ 3× 200のフィルターを 20 枚使用し，ストライド 1 で畳み込みが行われる．その後の処理は皮肉検出 CNN と同様である．全結合層ノード数は 1 層目から，20, 1000, 450, 2 である．出力層では softmax 関数を使用し，皮肉である確率と非皮肉である確率の 2 つが出力される．

（2）皮肉検出データセット 2 の CNN

皮肉検出データセット 2 を用いた実験で使用した CNN は，1 層の畳み込み層，1 層のプーリング層，4 層の全結合層からなる．サイズ 3× 200のフィルターを 88 枚使用し，ストライド 1 で畳み込みが行われる．その後の処理は皮肉検出 CNN と同様である．全結合層のノード数は 1 層目から，88, 650, 600, 2 である．出力層では softmax 関数を使用し，皮肉である確率と非皮肉である確率の 2 つが出力される．

4.2.2．RNN（Recurrent Neural Network）

RNN はネットワークに回帰結合を持ち，時系列などの系列データを学習することができるネットワークである．皮肉検出タスクにおける CNN の有効性を確認するため，RNN による皮肉検出を行う．ネットワーク構造およびハイパーパラメーターは，皮肉検出 CNN と同様に Optuna を用いてデータセットごとに最適化した．

（1）皮肉検出データセット 1 の RNN

皮肉検出データセット 1 では 3 層の RNN を使用した．ノード数は，1 層目から 200, 150, 2 である．入力データは CNN による皮肉検出と同じもの（図 2）を使用し，横方向に展開した単語分散表現が時刻ごとに順番に入力さ

表 4　皮肉検出データセット 1 に対する評価

	正解率	適合率	再現率	F 値
皮肉検出 CNN	0.823	0.817	0.834	0.825
CNN	0.758	0.781	0.719	0.748
RNN	0.738	0.737	0.744	0.739
LSTM	0.743	0.747	0.742	0.742

表 5　皮肉検出データセット 2 に対する評価

	正解率	適合率	再現率	F 値
皮肉検出 CNN	0.815	0.816	0.815	0.815
CNN	0.799	0.819	0.769	0.793
RNN	0.657	0.664	0.704	0.668
LSTM	0,776	0.792	0.753	0.771

れる．そのため入力層のノード数は 200 である．出力層では，softmax 関数によって皮肉である確率と非皮肉である確率の 2 つが出力される．

（2）皮肉検出データセット 2 の RNN

皮肉検出データセット 2 では 4 層の RNN を使用した．ノード数は，1 層目から 200, 350, 100, 2 である．その他は(1)と同様である．

４.２.３．LSTM（Long Short-Term Memory）

RNN は時系列を学習できるという特徴を持つが，学習する系列が長いほど損失関数の勾配を計算する際に勾配消失問題，勾配爆発問題が起きやすいという問題点がある．このうち勾配消失問題を解決するために考案されたのが LSTM である．LSTM は，情報を保持する記憶セルと，記憶セルの情報を取捨選択するゲートと呼ばれる仕組みによって，長期的な系列の処理が可能である．皮肉検出タスクにおける CNN の有効性を確認するためのもう 1 つの指標とするため，LSTM による皮肉検出を行う．ネットワーク構造およびハイパーパラメーターは，皮肉検出 CNN と同様に Optuna を用いてデータセットごとに最適化した．

（1）皮肉検出データセット 1 の LSTM

皮肉検出データセット 1 では 6 層の LSTM を使用した．ノード数は 1 層目から 200, 100, 500, 600, 100, 2 である．入力データは RNN 同様，CNN による皮肉検出と同じもの（図 2）を使用し，横方向に展開した単語分散表現が時刻ごとに順番に入力される．そのため入力層のノード数は 200 である．出力層では softmax 関数によって皮肉である確率と非皮肉である確率の 2 つが出力される．

（1）皮肉検出データセット 2 の LSTM

皮肉検出データセット 2 では 7 層の LSTM を使用した．ノード数は 1 層目から 200, 50, 200, 250, 250, 600, 2 である．その他は(1)と同様である．

４.２.４．皮肉検出の結果

皮肉検出 CNN は事前学習した感情分析 CNN の学習済みパラメーターを使用し，皮肉検出データセットのトレーニングデータでファインチューニングした．その他の比較実験用ネットワーク（CNN，RNN，LSTM）はパラメーターをランダムな数値で初期化して，同じトレーニングデータで学習を行った．皮肉検出 CNN および比較実験用ネットワークの皮肉検出データセット 1 のテストデータに対する正解率，再現率，F 値を表 4 に，皮肉検出データセット 2 のテストデータに対する正解率，再現率，F 値を表 5 に示す．学習は 10 回行い，数値はその平均値である．

５．考察

まず表 4 と表 5 から，提案手法はデータセット 1 と 2，どちらのデータセットに対しても CNN の評価を上回っていることがわかる．このことから，感情の特徴を捉えたパラメーターを皮肉検出タスクでファインチューニングすることが，皮肉の検出に有効であることがわかる．また先行研究[5]では、LSTM と Attention 機構[18]を組み合わせた識別モデルによる実験が総データ数 42,000 件で行われおり、その精度は F 値 0.772 であった。それに加えて、データ中に出現する「上司」と「部下」といった名詞ペアの関係ベクトルを事前に作成し、モデルの追加入力として皮肉検出を行う実験も行われており、その精度は F 値 0.802 であった。学習に用いたデータ数に大きな差がある中で、提案手法がこれら先行研究[5]の評価を上回ったことからも、感情の特徴を捉えたパラメーターのファインチューニングは皮肉の検出に有効であるとわかる。ファインチューニングによって精度向上を図る際，事前学習のタスクと目的タスクにはある程度の関連度が求められる．そのため，本研究で「皮肉は感情に起因する」とした仮定は妥当であると言える．また多くの場合，事前学習のタスクの精度が目的タスクの精度に大きく影響することが先行研究[19]で示されている．そのため，CNN による感情分析の精度を上げることで，CNN による皮肉検出の精度向上を同時に図ることができると考えられる．

次に，皮肉検出 CNN と CNN が，RNN と LSTM の評価を上回っていることがわかる．CNN 系ネットワークのネットワーク構造チューニングにおいて，畳み込みフィルターの縦幅を 3,5,7 の 3 種類でチューニングしたところ，3 が最も高い精度につながる結果となった．このことから，皮肉検出においては，RNN 系ネットワークのように系列全体の特徴を学習するよりも，単語の位置関係の特徴を細く学習する方が有効なのではないかと考えられる．

ノイズとなりうるテキストの除去を行ったデータセット 1 に対する実験（表 4）では，CNN，RNN，LSTM の 3 つのモデルの評価に統計的に有意な差はなく，同等の結果となった．一方で，ノイズの除去を行っていないデータセット 2 に対する実験（表 5）では，RNN が大きく評価を落とす結果となった．CNN は，畳み込みによって入力されたテキスト中の単語の関係に関する特徴を抽出し，

プーリングで抽象化することである程度のノイズを無視することができる．LSTM は単語系列のパターンを学習するが，ゲートと呼ばれる情報の取捨選択機構あるため，ノイズに対して堅固である．しかし，RNN にはゲートが存在せず，単語系列のパターンのみを学習するため，データセット2に対する実験ではノイズの影響を大きく受け，評価が落ちたと考えられる．

６．むすび

本研究では，「皮肉は感情に起因する」と仮定し，まずテキストのネガティブとポジティブを識別する感情分析 CNN の事前学習を行った．そして，この事前学習で得られた感情の特徴を捉えたパラメーターを，皮肉検出 CNN でファインチューニングすることで皮肉の識別を行う手法について検証した．実験は，ファインチューニングの有無とニューラルネットワークの種類，さらにデータセットの環境という 3 つの観点から行い，提案手法の有効性を確認した．

トレンドである BERT との比較実験を同じデータセットの下で行う必要がある．諏訪らの研究[6]は，BERT を用いて F 値 0.82 の皮肉検出精度を出しており，これは皮肉検出 CNN のデータセット 1 に対する評価と同程度である．しかし諏訪らは，クラウドソーシングを用いた複数人でのテキストのアノテーションによって，より信頼性が高く実験に理想的な皮肉検出データセットを構築し実験に使用しているため，ノイズのある実環境で BERT がどの程度堅実に精度を出せるかは不明である．BERT との比較を正確に行うためには，同じデータセットで実験を行う必要がある．

次に，テキストの前処理の最適化を行う．例えば，本研究ではテキストを数値化するための手法として単語分散表現を用いた．そのため，感情分析と皮肉検出のどちらのタスクも単語分散表現の質に依存することになる．その他にもテキスト中の品詞の取捨選択，正規化のルール作成など，前処理段階の最適化を行うことで精度向上を目指す．

続いて，皮肉検出に有効な新たな特徴量の発見である．先行研究[7]では，「嬉しい」，「悲しい」，「怒り」といった細かい感情や，話し手のパーソナリティを識別する別ネットワークの出力を皮肉検出ネットワークに活用する手法が提案されている．このような有効な特徴量を見つけることも今後の課題とする．

最後に，会話の流れを考慮した皮肉の検出を行うことも今後の課題である．本研究は独立した文章から皮肉を検出することを目的とした．しかし，皮肉には 2 人以上の話し手によって成り立つものも存在する．例えば，

A：「私の論文はどうでしたか？」
B：「驚くほどよく眠れたよ．」

というような会話である．会話であれば，そこには会話の流れが発生し，皮肉も会話の流れを前提とした内容になる．皮肉の検出に関する研究を進めるにあたって，今後は会話の流れを考慮した手法についても検討したい．

謝辞

本研究を進めるにあたり，熱心なご指導と適切なご助言を賜りました修士論文指導教員の平原誠准教授に心から感謝の意を表します．

参考文献

1) 肥合智史，嶋田和孝，“皮肉検出のための皮肉の対象の推定，”信学技報，vol.118, no.210, pp.7-11, 2018.
2) 徐 続非，肥合智史，嶋田和孝，“日本語 Wikipedia エンティティベクトルを外部知識を利用した皮肉検出，”信学技報，vol.120, no.374, pp.25-30, 2021.
3) 魚住惟，内田ゆず，荒木健治，“皮肉検出における感情生起要因の有効性，”第 17 回情報技術フォーラム，pp.163-166, 2018.
4) 肥合智史，嶋田和孝，“評価表現に着目した皮肉文の分析と抽出，”火の国情報シンポジウム 2016, no.4A-1, 2016.
5) 肥合智史，嶋田和孝，“関係ベクトルを利用した皮肉の検出，”言語処理学会 第 24 回年次大会 発表論文集，pp.829-832，2018.
6) 諏訪 幸輔，張 建偉，“BERT 及び絵文字を利用した日本語文における皮肉の検出，”第 13 回データ工学と情報マネジメントに関するフォーラム，no.H31-4, 2021.
7) S. Poria, E. Cambria, D. Hazarika, and P. Vij, “A deeper look into sarcastic tweets using deep convolutional neural networks,” COLING, pp.1601-1612, 2016.
8) T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” Workshop at ICLA, 2013.
9) T. Kudo, K. Yamamoto, and Y. Matsumoto, “Applying conditional random fields to Japanese morphological analysis,” Proc. EMNLP, pp.230-237, 2004.
10) “Gensim”, https://radimrehurek.com/gensim/#, 2020 年 1 月 20 日
11) 猪原 敬介，内海 彰，“日本語類似度・関連度データセットの作成，”言語処理学会 第 24 回年次大会 発表論文集， pp.1011-1014，2018.
12) 鈴木正敏，松田耕史，関根聡，岡崎直観，乾健太郎，“Wikipedia 記事に対する拡張固有表現ラベルの多重付与，”言語処理学会 第 22 回年次大会，pp.797-800，2016.
13) M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” Proc. NAACL, vol.1, pp.2227-2237, 2018.
14) J. Howard, and S. Ruder, “Universal language model fine-

tuning for text classification," Proc. ACL, vol.1, pp.328-339, 2018.
15) Y. Suzuki, "Filtering Method for Twitter Streaming Data Using Human-in-the-Loop Machine Learning," JIP, vol.27, pp.404-410, 2019.
16) T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna:a next-generation hyperparameter optimization framework," Proc. SIGKDD, pp. 2623-2631, 2019.
17) "TwitterAPI", https://developer.twitter.com/en/products/twitter-api, 2020 年 3 月 16 日
18) D. Bahdanau, K. Cho, and Y. Bengio, "Neural ma- chine translation by jointly learning to align and translate", Proc. ICLR, 2015.
19) W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, and M. Zhou, "ProphetNet:predicting future n-gram for sequence-to-sequence pre-training, " Proc. EMNLP, pp.2401-2410, 2020.