自然言語処理によるフェイクニュース判断の自動化

発表者: 総合情報学科 メディア情報学 コース 学籍番号 1510151 柳 裕太
指導教員: 田原 康之 准教授

1 序論

昨今の SNS の普及により，誰もが情報を発信・収集できるようになった．特に最近ではテキストのみならず，画像や動画と併せて情報の発信が可能である．一般論として，テキスト単体と比べて画像や動画と併せて発信されたマルチメディア情報の方が多くの注目を得やすい．逆にこれを利用して，故意に情報を捏造して発信することによって人々を誤った方向へ扇動するフェイクニュースも存在する．フェイクニュースが広まると、大規模なマイナスの影響が出る可能性があり、場合によっては重要な公共の出来事に影響を及ぼしたり、操作したりすることさえある．例えば 2016 年の米国大統領選では，2 名の候補者を支持させるためのフェイクニュースが多く拡散され，とりわけ Facebook 上では 3700 万回以上共有された [1].

虚偽の情報ながら，扇動ではなく皮肉や風刺を込めたジョークニュースも存在する．有名な発信メディアとしては，英語では the Onion，日本語では虚構新聞が該当する．あくまで扇動ではなく笑いを提供するためのものであり，多くの場合それは批判の的にはなりにくかった．しかしながら，ジョークニュースはフェイクニュースと同じく限りなく真実を模した形式をとるため，昨今では同じく SNS 上で拡散されやすく，同時に批判に晒されることもある．

当研究では，扇動のために故意に情報を捏造して発信された情報をフェイクニュース，事実を発信した情報を正しいニュース，そして風刺や皮肉を込めて発信された情報をジョークニュースとして定義する．

フェイクニュースに限らず，風評や web ページの信憑性を評価するモデルの構築の研究は数多く行われており，特に機械学習による分類が非常に盛んに行われている．なかでも Granik らの研究 [6] や Gilda の研究 [5]，そして松尾の研究 [25] により，単語埋め込みとナイーブベイズ分類器や SVM，決定木といった教師あり学習を組み合わせることによって，フェイクニュースや流言を分類するタスクで優秀な分類成果を挙げることが報告された．ほかにも Wu らの研究 [24] によると，SNS 上で拡散された情報に対して，“誰が・どのような経緯で拡散したか” という情報から信憑性を判断するモデルも提案された．Rubin らの研究 [17] によれば，正しいニュース・ジョークニュースの分類にも機械学習によるアプローチが有効であることが示されていた．正しいニュース・フェイクニュース・ジョークニュースの 3 カテゴリ分類においても研究が行われている．特に Horne と Sibel の研究 [10] によると，フェイクニュースは正しいニュースよりジョークニュースに近い性質をもち，真実に近い形式をとるほど高い説得力をもつことが示されていた．

上記の機械学習を使った研究では，いずれもテキストのみの情報を対象としていた．別の対象として，テキスト・画像を併せた情報を分類する機械学習モデルの検討も数多く行われている．大まかな形としては，まずテキスト・画像を何らかの方法でベクトル化する．その後 2 種のベクトルを結合し，真偽判定を行うモデルに渡す形をとっている．例えば Jin らの研究 [11] では，テキストでは LSTM，画像では VGG19 を使用してベクトル化しており，それに加えて Attention とソーシャルコンテキスト (ハッシュタグ，URL 等) により更に高精度な分類を行うモデルが提案されていた．また Wang らの研究 [23] では，EANN というモデルが提案されている．これは画像のベクトル化においては同じく VGG19 を使用しているが，テキストではテキスト CNN を使用していた．

上記の EANN モデルのような画像・テキスト双方を扱うモデルでは，実際に真実・フェイクとのカテゴリ分類において画像単独・テキスト単独の分類に比べて優秀な成績を収めていた [23]. しかしながら，あくまで “真実なのかそうでないのか” という 2 カテゴリで分類しているため，“他者を欺くための情報なのか，皮肉・風刺を込めた情報なのか” という観点での分析がなされていない．

本研究では，画像つきで発信された情報に対して，正しい情報か・フェイクニュースか・ジョークニュースかを判断するモデルを構築する．このモデルを使い，従来から画像・テキスト複合のデータセットに対して 3 カテゴリでも優秀な分類が行えることを示すことを目指す．それにより，SNS ユーザの情報収集を支援するエージェントの開発につなげることが可能となる．

上記の提案する情報分類システムを検証するために，事前に用意されたデータセットを用いて 10 分割交差検定によって分析を行った．また上記システムの分類性能を評価するために，画像・テキスト単独で分類を行った結果と比較することで，提案システムが目標に適していることを示すことを目指した．その結果テキスト単独でのマクロ F 値が約 0.30，画像単独でのマクロ F 値が約 0.47 であったのに比べ，提案モデルのマクロ F 値は約 0.93 という数値を出し，提案モデルの有効性が示された．

本論文の構成は次の通りである．第 2 章では，本研究と関連のある研究や取り組みを紹介する．第 3 章では，本研究の目的と対象とする投稿を例とともに示す．第 4 章では，本研究が提案する手法を理論式と図とともに説明する．第 5 章では，本研究の性能を試すために実際に評価実験を行ったため，その詳細を記述する．第 6 章では，第 5 章で行った実験の結果を詳細に評価している．考察とともに，手法や結果から浮かび上がった課題についても指摘する．第 7 章では，本論文をまとめるとともに，今後の発展形についていくつかの展開を記載している．

2 関連研究

2.1 web ページの体裁による分類

福島らの研究 [26] では，web ページの体裁から信憑性を評価するモデルが提案された．これは，情報そのものではなく情報が掲載されている web ページがどういった形式やコンテンツを持っているかをアンケート調査によって判断するものであった．例えば，管理者の連絡元を表記したり，記事の公開・更新日時が明記されていたりしていた場合は，掲載情報の信憑性にポジティブな影響を与えていることが確認された．逆に，掲載リンクが機能していなかったり，広告が 1 つ以上表示されていたりしていた場合は，掲載情報の信憑性にネガティブな影響を与えることが確認された．

2.2 画像・文章の分析

画像分類のは近年目まぐるしい発展を遂げた．特に画像の被写体から分類するタスクにおいては，VGG19 のように 16-19 層の畳み込み層を取り入れたモデルが非常に高い分類成果を挙げることが報告 [18] された．また，VGG19 を含めた多くのモデルでは，事前学習済みモデルが配布されているため，自分で転移学習を行うことも容易である．

フェイクニュースに限らず，文章を対象とした分類はいくつかのタスクがある．例えば，文章から執筆者の感情を判断するセンチメント分析や，ニュース記事から該当するカテゴリを判断するカテゴリ分類などがある．当研究では，分類先のカテゴリが 3 種類に固定されているため，カテゴリ分類の一環とみなすことができる．機械学習によって分類する場合，第 1 章の通り非常に数多くの手法が使われてきた．最近では，ニューラルネットワークを活用して人間の短期記憶を再現した LSTM[8] では，人間の短期記憶を再現することによって，分類のみならず文章を生成するタスクにおいても発展している．また，上記では GPU による並列実行が難しいため，画像と同じく並列実行が可能な CNN をテキスト用にアレンジしたテキスト CNN も提案 [13] され，広く使われている．

画像と文章を組み合わせた研究も数多くなされてきた．例えば，画像を CNN で分析して LSTM によってキャプションを生成する研究 [21] によって，より精度の高いキャプション生成ができたことが報告された．キャプション生成のほかに，画像に対して文章で視覚質問 (画像に写ったものを問う) に応答することを目的とした VQA[2] というモデルも提案された．

2.3 フェイクニュース対策

現在，フェイクニュースを判断する手法の 1 つに有識者によって事実関係を確認するファクトチェックがある．例えば Politifact.com では Truth-o-meter という独自指標によって，政治的主張に対して疑わしさを 7 段階で評価 [9] している．その中では，真実ではあるが重要な部分を省くことによて誤解を招きやすい “half-true” や，一部真実を含むものの，重要な事実が無視されていることを示す “mostly-false”, 主張が正確ではない “false”，そして完全に虚偽であり，ばかげた主張とする “pants-on-fire” など，多くの評価名が用意されている．

フェイクニュース自体への対策が発展していく中で，フェイクニュースを “真実か嘘か” という基準から判断すること自体に疑義を唱える取り組みも存在する．現在，Mike Tamir 氏によって立ち上げられた FakerFact という web アプリケーションがある [20]. この取組では，フェイクニュースはセンセーショナルな文章を書くことによって読者の本能に働きかけ，読者に拡散させる扇動を目的としていることに着目していた．この web サイトでは Walt という独自の AI を搭載しており，文章を “真実か嘘か” は判断せず，文章の論調から以下の 6 カテゴリに分類していた．

- Journalism: 事実をベースとした文章
- Wiki: 辞書的文章
- Agenda Driven: 何かしらの意図がみられる文章

- Opinion: 意見が書かれた文章
- Sensational: 扇動を目的とした文章
- Satire: 風刺・皮肉

あくまで真偽は判断せず読者に何を伝えたいのかを類推することで，読者が真偽を判断する手助けになることがこのモデルの目的であった．このモデルでも，Satire として風刺・皮肉をもつ文章 (ジョークニュース) が区別されてあった．

このようにフェイクニュースを判断するにあたって，近年では “真実か嘘か” という観点にとらわれない多くのアプローチや分類が行われていることがわかる．

3 研究目的

3.1 分類対象

今回対象とする情報は，SNS 上で投稿された画像つきで発信されたニュースである．そのなかでも，正しいニュースを発信していたもの，フェイクニュースを発信していたもの，ジョークニュースを発信していたものが対象となる．それぞれの例を今回扱ったデータセットから抜粋したものが以下の図 1 である．

いずれも 2013 年に発生したボストンマラソン爆弾テロ事件に関して Twitter 上で投稿されたものであった．図 1a は実際にボストン市傘下組織が作成した被疑者の情報を Chicago Sun-Times が Twitter に投稿したもの，図 1b はテロ後に Reddit や 4chan の有志によって実行犯の調査が行われた件に対して “bananas” と茶化すような言葉を投げかけているもの，図 1c は実際に上記掲示板上で実行犯の調査が行われた結果，全くの別人を槍玉に挙げているものである．

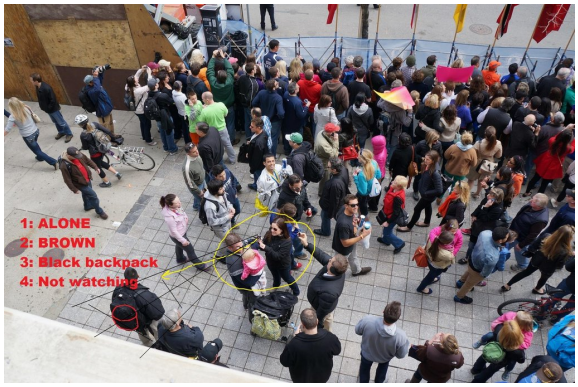
実際に，ボストンマラソン後ではインターネット上で盛んに犯人探しが行われた結果，事件前に行方不明になっていたスニル・トリパティ（Sunil Tripathi）さんが犯人として扱われ，更にその後一般報道メディアによってトリパティさんの家族に取材が行われるなど，フェイクニュースが実害として現実になった [7]. これを受け，Reddit では実際に犯人探しの過熱で無関係の個人とその家族に迷惑をかけたとして謝罪した [15].

3.2 達成目標

当研究では，上記対象を正確に 3 カテゴリへ分類するモデルを構築することを目標としている．具体的には，入力として画像と文章を持ち，それに対してどのカテゴリが該当するかを出力するモデルとなる．ジョークニュースとフェイクニュースを完全に区別することで，センセーショナリズムによる影響を最小限に留め，なおかつ高い精度を維持することを目指すことにする．

当研究を更に発展させると，SNS 上でフェイクニュースに該当する記事に対してユーザへ警告を出したり，ジョーク記事の場合はそれを知らせる追加情報を与えたりするユーザエージェントを開発することへ繋げられる．また繰り返しフェイクニュースを発信するユーザがいる場合は，

(a) Boston RIC released this flier showing at large suspect Dzhokhar Tsarnaev. He may be armed & dangerous



(b) The detail of these photos used to identify the Boston Marathon bombing suspect is bananas…

(c) Reddit is on to something... Boston Bomber #2 sure look

図 1: 当研究で扱う 3 カテゴリの投稿例: (a) 正しいニュース，(b) ジョークニュース，(c) フェイクニュース

運営側へアカウント停止等のペナルティを迅速に進言するエージェント開発にも発展可能である．

4　提案手法

4.1　モデル概観

この章では，提案モデルがもつ複合特徴量抽出器とニュース分類器について紹介する．その後にこの 2 要素を統合して転移学習が可能な表現を学習する方法について説明する．今回提案したモデルは，以下の図 2 の通りである．

提案モデルの目的は，画像と文章で発信された情報に対して，正しいニュースか・フェイクニュースか・ジョークニュースかを分類するために，必要な特徴表現を学習することであった．提案モデルは複合特徴量抽出器とニュース分類器の大きく 2 部分に分けることができた．まず複合特徴量抽出器は，今回扱う情報が文章と画像を含むため，各メディアに対して特徴化する抽出器があった．その後それぞれの特徴を 1 つに連結し，複合特徴を形成した．複合特徴はニュース分類器に送られ，最終的には 3 カテゴリのどれに該当するかが判断された．

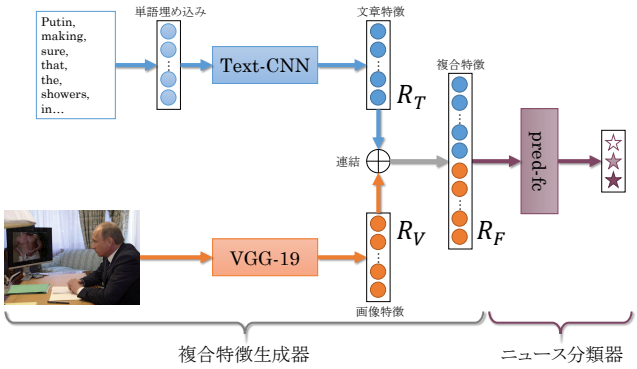


図 2: 提案モデル図．青: 文章特徴量抽出器，橙: 画像特徴抽出器，紫: ニュース分類器．

4.2　複合特徴抽出器

4.2.1　文章特徴

文章特徴は，入力に英語の投稿をスペース毎に分割した英単語の連続リストをもった．まずは単語を単語埋め込みでベクトル化した．その後単語の羅列から分類に有効な情報を得るために，文章特徴を抽出する核として CNN (convolutional neural networks: 畳み込みニューラルネットワーク) を採用した．CNN はコンピュータビジョンやテキスト分類などの多くの分野で効果的であることが示されていた [4, 12]. 図 2 の通り，提案手法では CNN の発展形であるテキスト CNN(Text-CNN)[13] を採用した．テキスト CNN の構造は図 3 の通りである．複数のウィンドウで畳み込むことで，様々な角度から特徴を抽出することを実現した．

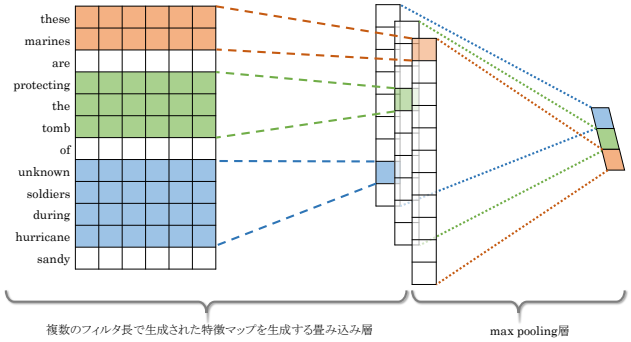


図 3: テキスト CNN の図．Wang らの研究 [23] を参考に作成．

具体的な手法では，EANN が採用したテキスト CNN と同じ流れを汲み [23], 最終の全結合層の隠れ層に独自に dropout を採用した形をとった．今回使用した文章特徴抽出器の理論式を以下に引用する．

投稿の i 番目の単語を k 次元の単語埋め込みに変換する際に抽出された単語埋め込みベクトルを $T_i \in \mathbb{R}^k$ とする．このとき，n 単語から抽出された投稿は以下の式 1 で表現ができる．

$$T_{1:n} = T_1 \oplus T_2 \oplus \ldots \oplus T_n, \tag{1}$$

$\oplus$ はベクトルを連結 (concatenation) することを意味する記号である．ここで単語埋め込み化された投稿は，畳み込み層へ送られる．畳み込み層では h 単語分のウィンドウサイズがある．これは単語埋め込み化された投稿から連続して取り出す単語埋め込みベクトルの数を意味する．i 番目を基準に h 単語分取り出された場合，フィルタ時の処理は以下の式 2 の通りである．

$$t_i = \sigma(W_c \cdot T_{i:i+h-1}). \tag{2}$$

この $\sigma(\cdot)$ は活性化関数の 1 つである ReLU(ランプ関数) を表し，W_c はフィルタの重みを意味する．式 2 が投稿内で適用されると，式 3 の通り 1 つの発言に対し 1 つの特徴ベクトルが得られる．

$$t = [t_1, t_2, \ldots, t_{n-h+1}]. \tag{3}$$

この t ベクトルに対して最も重要な特徴を得るために，ベクトル内で最大値の要素のみを取り出す max-pooling が行われる．

これで 1 つのウィンドウサイズから 1 つの特徴が得られるが，多くの粒度の特徴を得るために当手法では複数のウィンドウサイズから複数の値を取得している．特定のウィンドウサイズに着目すると，n_h 分異なるフィルタが存在することになる．もし使用可能なウィンドウサイズが c 存在する場合，全体で $c \cdot n_h$ だけフィルタが存在することとなる．投稿から各ウィンドウサイズで max-pooling までされた文章特徴は $R_{T_c} \in \mathbb{R}^{c \cdot n_h}$ と表現できる．

max-pooling を終えた文章特徴は全結合層に渡され，最終的に式 4 によって画像特徴抽出器が出力する特徴ベクトルの次元 (p とする) に合わせたテキスト特徴 $R_T \in \mathbb{R}^p$ となる．

$$R_T = \sigma(W_{tf} \cdot R_{t_c}), \tag{4}$$

ここで W_{tf} は全結合層における重みを意味する．なお，この全結合層では隠れ層に dropout を当研究では採用した．dropout は Hinton らによって提案された手法 [19] で，学習時に指定された確率で無作為に W_{tf} 内の要素を無効化 (0 に) してモデルの自由度を制限することで，モデルが訓練データセットに特化しすぎて汎用性が失われる過学習に繋がりにくくなる利点が報告されたものである．

4.2.2 画像特徴

画像から効率的に特徴を抽出するために，当研究では事前学習済みの VGG19[18] を起用した．VGG19 は畳み込み 16 層と全結合層 3 層から形成され，最終的には 1000 次元の特徴ベクトルが出力される．当研究では最終の全結合層のみ改変し，文章特徴のベクトル次元数と同じ数の次元をもつベクトルを出力するようにした．また改変した最終全結合層以外は，過学習を防ぐために事前学習の状態を維持することにした．以下に，一部を改変した VGG19 を利用したモデルの理論式を記す．第 4.2.1 節で記したように，最終的な特徴ベクトルの次元数は p とする．VGG19 では畳み込み 16 層で $7 \times 7 \times 512$ の行列となり，その後 2 層の全結合層によって $1 \times 1 \times 4096$ に整形され，最終第三全結合層 (fc19) によって $1 \times 1 \times 1000$ の画像特徴ベクトルを出力する．Wang らの研究 [23] では VGG の fc19 の出力を p に整形していたが，当研究では直接 fc19 を改変して $1 \times 1 \times p$ の画像特徴ベクトルを出力するようにした．この改変した fc19 によって算出される画像特徴 $R_V \in \mathbb{R}^p$ は以下の式 5 の通りである．

$$R_V = \sigma(W_{vf} \cdot R_{V_{\mathrm{VGGfc18}}}), \tag{5}$$

$R_{V_{\mathrm{VGGfc18}}}$ は VGG19 の第 18 層である全結合層が出力した $1 \times 1 \times 4096$ ベクトルである．

こうして文章特徴・画像特徴が抽出され，最終的には 2 つの特徴ベクトルを 1 つに連結したものが複合特徴である．理論式で表記すると，文章特徴 R_T と画像特徴 R_V が 1 つに結合されるため，複合特徴 R_F は以下の式 6 によって表現できる．

$$R_F = R_T \oplus R_V \in \mathbb{R}^{2p}. \tag{6}$$

以降においては，複合特徴抽出器全体を表現するときは $G_f(M; \theta_f)$ と表現することにする．M は複合特徴抽出器へ入力される投稿，θ_f は学習対象となるパラメータを意味する．

4.3 ニュース分類器

複合特徴はニュース分類器 (図 2 内 ‘pred-fc’ が該当) にて正しいニュース・フェイクニュース・ジョークニュースとして分類された．具体的には隠れ層を含む全結合層と softmax から形成され，最終的な分類が行われた．この部分の理論式は以下の通りである．入力となる複合特徴は R_F であるとき，ニュース分類器は $G_f(\cdot; \theta_d)$ と表現することにする．ここで θ_d はニュース分類器内で学習対象となるパラメータを示す．投稿全体に対して i 番目の投稿を m_i とするとき，m_i がフェイクニュースもしくはジョークニュースである確率は以下の式によって算出される．

$$P_\theta(m_i) = G_d(G_f(M; \theta_f); \theta_d). \tag{7}$$

モデルの目的は自動で正確に正しいニュース・フェイクニュース・ジョークニュースを分類することである．そのため正解ラベルとして Y_d を使用して，以下の式によってクロスエントロピー誤差を損失として算出する．

$$L_d(\theta_f, \theta_d) = -\mathbb{E}_{(m,y)\sim(M,Y_d)}\left[\sum_{k=0}^{2} y_k \log P_\theta(m)\right]. \tag{8}$$

最後に，当研究がベースとした Wang らの研究では確率的勾配降下法 (SGD: Stochastic Gradient Descent) によってパラメータを更新していたが，当研究では 2015 年に Diederik P. Kingma らが提唱した Adam という手法 [14] を用いてパラメータを更新することにした．

5 評価実験

5.1 データセット

今回の実験での訓練データセットでは，Boididou らの研究 [3] によって提案された Twitter 投稿データセットを使用した．こちらも Twitter 上でフェイクニュースを検出するために作られたデータセットであるが，付加されたラベルとして Real, Fake, そして Humor があり，ジョーク

ニュースを含めた 3 カテゴリ分類に適したものとなっているため，当研究で採用した．データセットでは訓練用と検証用として 2 部に分かれていたが，当研究では訓練用とされた部分を対象に 10 分割交差検定することにした．データセット内ではツイート文章と画像のみならず，タイムスタンプや投稿者といったソーシャルコンテキスト情報も含まれている．当研究ではソーシャルコンテキストは対象に含まず，文章と画像のみで 3 カテゴリ分類することを目指した．

5.2 比較対象手法

今回，画像つき文章投稿を 3 カテゴリに分類する提案手法の有効性を調べるために 2 種類の比較対象手法を用意した．1 つは文章のみで投稿を分類する手法 (以降，Text と表記)，もう 1 つは画像のみで投稿を分類する手法 (以降，Image と表記) であった．いずれも上記提案モデルから文章・画像特徴抽出器を除外したモデルを使用した．また Text は入力データを提案モデルが使用したデータセットから画像を削除したものを使用した．Image は全投稿で使用された画像を対象とし，同じ画像に対して複数の文章投稿があった場合は 1 件として数えることにした．

5.3 実験条件

5.3.1 Text

モデルに入力する前に，単語埋め込みへの変換の事前処理として，スムーズに単語埋め込みに変換できるようにするために，投稿からハッシュタグ (#) のような記号を除去し，全ての大文字を小文字に変換してスペースで分割した．投稿から分割された各単語を単語埋め込みに変換する際は，Google News データセットから事前学習済みの word2vec モデル [16] を使用した．このモデルでは，各単語を 300 次元のベクトルに変換するものであった．ここで wod2vec モデルに該当しない単語が出現した場合，<unknown>として seed 値固定ランダムベクトルを生成することにした．また，投稿の全単語中 50% 以上が<unknown>の場合，実態に則さない学習を避けるために学習対象から外すことにした．その後テキスト CNN に送られ，1 つの文章に対して 1 つの 300 次元の特徴ベクトルが抽出され，ニュース分類器に渡す形となった．なお，ウィンドウサイズは 2-5 までの 4 種類を用意し，全結合層の隠れ層は 1 つ用意し，ユニット数は 300 とした．隠れ層では 50% のユニットが無視される Dropout を導入した．ニュース分類器内でも隠れ層は 1 つ用意し，ユニット数は 300，上記と同じ条件の Dropout も導入した．検証にあたり最大エポック数は 100 とし，過学習の兆しが見えたら学習を打ち切る Early stopping も導入した．

5.3.2 Image

モデルの複雑化を避けるため，1 つの投稿に複数枚画像が付加されていた場合は最初の 1 枚のみをモデルに入力させることにした．画像は事前学習済み VGG19 モデルに入力し，1 つの画像に対して 1 つの 300 次元のベクトルが抽出され，ニュース分類器に渡す形となった．本来の VGG19 は最終層にて 1000 次元のベクトルを出力するが，最終層のみ改変して 300 次元のベクトルが出力されるようにした．また前記の通り過学習を避けるために最終層を除き事前学習済みの状態を維持させることにした．ニュース分類器とエポック数，そして Early stopping の条件は上記 Text と同一であった．

5.3.3 提案手法

提案手法では，Text と Image を統合した形をとったため，画像・文章の部分は上記と同様の条件をとった．画像・文章の特徴を結合するため，ニュース分類器に渡されるのは 600 次元のベクトルであった．それにあわせ，隠れ層のユニット数も 600 とした．

5.3.4 使用データ統計

上記の条件を踏まえ，提案手法・Text・Image が扱う 3 カテゴリの投稿件数は以下の表 1 の通りである．Text が使用するデータは提案手法が扱うデータから画像を削除したものであるため，提案手法と全く同じ件数になった．Image は，同じ画像に対して複数の投稿があったため，他 2 手法と比べて非常に少なくなっている．

表 1: 提案手法と比較対象手法が扱うカテゴリ毎の投稿数

手法	Real	Fake	Humor
Text	3021	4233	1509
Image	172	157	82
提案手法	3021	4233	1509

5.3.5 評価指標

評価指標では，Precision(精度), Recall(再現率), F 値 (左 2 値の調和平均) を使用することにした．算出する方法上各カテゴリ毎に上記指標があるが，今回使用するデータセットでは極端にカテゴリが偏っていないので，各カテゴリの指標を先に算出してから 3 カテゴリの平均をとるマクロ平均を評価に使うことにした．

5.4 実験結果

3 モデルに対して 10 分割交差検定を行った結果が以下の表 2 の通りである．

表 2: 各モデルの分類成果 (マクロ平均)

手法	Precision	Recall	F 値
Text	0.3649	0.3677	0.3016
Image	0.4942	0.5055	0.4667
提案手法	0.9268	0.9362	0.9286

この結果を見ると，提案手法が他 2 手法と比べて非常に高い分類成果を挙げたことが読み取れた．また，比較対象手法内で比べると画像単体の方が分類成果が良好である点もみられた．

6 評価

6.1 考察

今回の評価実験では，提案手法が 3 指標全てにおいて比較対象手法より優れた分類成績を収めた．これにより，SNS 上で画像つきの投稿を対象にした場合，正しいニュース・フェイクニュースの分類タスクのみならず，ジョークニュースも含めた分類においても従来のマルチメディアモデルのアプローチが有効であることが示唆されたのではないかと考えられる．

また比較対象手法に限って結果を観察すると，文章単体より画像単体の分類の方が優秀な分類成績であった．これは自然言語より画像の方が分類タスクにおいて研究が進んでいることや，SNS 上の投稿であった故に単語埋め込みに変換する際に<unknown>に変換されやすい傾向にあったことや，文章の場合英語以外の投稿に対応できないものの，画像においては英語圏以外の投稿であっても十分言語の違いに影響されにくかったことなど，いくつかの原因が推察される．

6.2 課題

今回分類するにあたり，大きな課題となったのが文章投稿の単語埋め込みへの変換であった．例えば今回使用したデータセットが Twitter から収集されたものであったため，事前学習済み word2vec モデルが対応できない短縮語や造語 (ハッシュタグなど) といったユーザ生成コンテンツに対応することが難しかった．

また，このモデルに限らずフェイクニュース検出というタスクにおいては，Wang らの研究 [23] によってある問題点が指摘されていた．訓練に使ったデータセットが扱うイベントや出来事の特殊性の影響を受けることにより，検証する時に訓練になかった別のイベントや出来事が使われた場合に正常な判断ができなくなる点であった．

さらに，このモデルは英語のみを対象としたものであった点も挙げられた．データセット内一部では他国の言語が含まれていたため，単語埋め込みに変換する際に大幅に<unknown>に変換される傾向もあった．

7 おわりに

7.1 本論文のまとめ

本研究では，SNS 上で画像と文章を併せて発信された情報に対して，正しいニュース・フェイクニュース・ジョークニュースを判断するモデルを提案した．実際に 3 カテゴリ分類を行った結果，文章・画像単体から分類した場合に比べて，全ての評価指標において非常に優秀な分類成績を挙げた．これにより SNS 上における画像つき投稿に対して，ジョークニュースを含めた 3 カテゴリ分類も有効であることが示された．

7.2 今後の展望

このモデルの発展形として，いくつかの方法が考えられる．

例えば文章特徴生成器に対して，テキスト CNN ではなく Vosoughi らの研究 [22] によって SNS 投稿を分析するために提案された，文字単位でベクトル変換する方法を採用することなどが考えられる．

また，データセットが扱う出来事やイベントによる特殊性の対策として，Wang らの研究 [23] では敵対的生成ネットワーク (GAN) を模倣する形をとることが挙げられていた．このイベントや出来事による特殊性を排するために，真偽分類に加えて扱われたイベントも分類することによって，特徴化する際に特殊性を排し，フェイクニュースの普遍的な特徴を抽出するようなアプローチが行われていた．実際にこれによって分類精度が改善した点が上記研究によって報告されていたため，当研究でも有効に働く可能性がある．

提案手法を日本語投稿に対応させることを考えた場合，まず SNS 上で日本語による画像つきの正しいニュース・フェイクニュース・ジョークニュースの投稿を収集する必要があると考えられる．その上に形態素解析によって分かち書きする部分を加え，さらに日本語用の事前学習済み word2vec を用意する必要だと考えられる．もしも既に日本語投稿による 3 カテゴリ分類済みのデータセットがあれ

ば投稿を収集する必要はないが，残念ながら国内に今回使用したデータセットに近い規模をもつものがないのが現状である．

参考文献

[1] Hunt Allcott and Matthew Gentzkow. “Social Media and Fake News in the 2016 Election”. In: *Journal of Economic Perspectives* 31.2 (May 2017), pp. 211–36. DOI: 10.1257/jep.31.2.211. URL: http://www.aeaweb.org/articles?id=10.1257/jep.31.2.211.

[2] S. Antol et al. “VQA: Visual Question Answering”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 2425–2433. DOI: 10.1109/ICCV.2015.279.

[3] Christina Boididou et al. “Verifying Multimedia Use at MediaEval 2015.” In: *MediaEval*. 2015.

[4] Ronan Collobert et al. “Natural language processing (almost) from scratch”. In: *Journal of Machine Learning Research* 12.Aug (2011), pp. 2493–2537.

[5] S. Gilda. “Evaluating machine learning algorithms for fake news detection”. In: *2017 IEEE 15th Student Conference on Research and Development (SCOReD)*. Dec. 2017, pp. 110–115. DOI: 10.1109/SCORED.2017.8305411.

[6] M. Granik and V. Mesyura. “Fake news detection using naive Bayes classifier”. In: *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. May 2017, pp. 900–903. DOI: 10.1109/UKRCON.2017.8100379.

[7] Lacey Gray. *Muslim Bashing in the Wake of Boston Bombing*. Apr. 2013. URL: https://news.nationalgeographic.com/news/2013/13/130426-boston-marathon-bombing-racism-hate-anti-arab-muslim-tamerlan-dzokhar-tsarnaev/ (visited on 02/01/2019).

[8] K. Greff et al. “LSTM: A Search Space Odyssey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (Oct. 2017), pp. 2222–2232. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2016.2582924.

[9] Angie Drobnic Holan. *The Principles of the Truth-O-Meter: How we fact-check*. Feb. 2018. URL: https://www.politifact.com/truth-o-meter/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i (visited on 01/24/2019).

[10] Benjamin D. Horne and Sibel Adali. “This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News”. In: *CoRR* abs/1703.09398 (2017). arXiv: 1703.09398. URL: http://arxiv.org/abs/1703.09398.

[11] Zhiwei Jin et al. “Multimodal Fusion with Recurrent Neural Networks for Rumor Detection on Microblogs”. In: *Proceedings of the 25th ACM International Conference on Multimedia*. MM ’17. Mountain View, California, USA: ACM, 2017, pp. 795–816. ISBN: 978-1-4503-4906-2. DOI: 10.1145/3123266.3123454. URL: http://doi.acm.org/10.1145/3123266.3123454.

[12] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. “A Convolutional Neural Network for Modelling Sentences”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (June 2014). URL: http://goo.gl/EsQCuC.

[13] Yoon Kim. “Convolutional Neural Networks for Sentence Classification”. In: *CoRR* abs/1408.5882 (2014). arXiv: 1408.5882. URL: http://arxiv.org/abs/1408.5882.

[14] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014). arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980.

[15] Sam Laird. *Reddit Apologizes for Boston Marathon ’Witch Hunt’*. Apr. 2013. URL: https://mashable.com/2013/04/22/reddit-apologizes-boston-marathon/ (visited on 02/01/2019).

[16] Tomas Mikolov and Ilya Sutskever. *Google Code Archive - Long-term storage for Google Code Project Hosting.* July 2013. URL: https://code.google.com/archive/p/word2vec/ (visited on 01/24/2019).

[17] Victoria Rubin et al. “Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News”. In: *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*. San Diego,

California: Association for Computational Linguistics, 2016, pp. 7–17. DOI: 10.18653/v1/W16-0802. URL: http://aclweb.org/anthology/W16-0802.

[18] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556 (2014). arXiv: 1409.1556. URL: http://arxiv.org/abs/1409.1556.

[19] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[20] Mike Tamir. *About FakerFact*. URL: https://www.fakerfact.org/about (visited on 01/24/2019).

[21] O. Vinyals et al. “Show and tell: A neural image caption generator”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 3156–3164. DOI: 10.1109/CVPR.2015.7298935.

[22] Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. “Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder”. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’16. Pisa, Italy: ACM, 2016, pp. 1041–1044. ISBN: 978-1-4503-4069-4. DOI: 10.1145/2911451.2914762. URL: http://doi.acm.org/10.1145/2911451.2914762.

[23] Yaqing Wang et al. “EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’18. London, United Kingdom: ACM, 2018, pp. 849–857. ISBN: 978-1-4503-5552-0. DOI: 10.1145/3219819.3219903. URL: http://doi.acm.org/10.1145/3219819.3219903.

[24] Liang Wu and Huan Liu. “Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate”. In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. WSDM ’18. Marina Del Rey, CA, USA: ACM, 2018, pp. 637–645. ISBN: 978-1-4503-5581-0. DOI: 10.1145/3159652.3159677. URL: http://doi.acm.org/10.1145/3159652.3159677.

[25] 松尾 省吾. “機械学習を用いた流言の検出に関する研究”. MA thesis. 電気通信大学院, 2018.

[26] 福島 隆寛 and 内海 彰. “Web ページの信頼性の自動推定”. In: 知能と情報 19.3 (2007), pp. 239–249. DOI: 10.3156/jsoft.19.239.