Convolutional Neural Network を用いた Fake News Challenge の検討

雨宮　佑基†　　酒井　哲也†

† 早稲田大学基幹理工学部情報理工学科　〒 169–8555 東京都新宿区大久保 3-4-1
E-mail: †yukiamemiya@fuji.waseda.jp, ††tetsuyasakai@acm.org

あらまし　Fake News は SNS を中心に拡散され，政治にまで大きな影響を及ぼす社会問題の 1 つである．その問題に対抗するため，機械学習や自然言語処理技術を用いてニュース記事の見出しと本文の関係を識別するタスクが，2017 年に Fake News Challenge Stage 1 として開催された．そのタスクにおいて最も高いスコアを獲得したチームは畳み込みニューラルネットワーク (CNN) を使用したが，その分類精度は低かった．そこで本研究では，CNN をベースとしたモデルを構築した上で階層分類を行い，その効果を検証する．また記事の見出しと本文の一致度合いだけでなく，不一致の度合いも測ることができる独自の評価方法も提案する．評価実験の結果，本研究で提案したモデルは，FNC-1 で使用された評価方法および本研究で提案された評価方法の両方において，設定されたベースラインより高い精度で分類を行った．したがって，このタスクに階層分類が有効であるということが確認できた．

キーワード　Fake News Challenge，スタンス検出，畳み込みニューラルネットワーク (CNN)，階層分類

1. はじめに

Fake News とは，マスメディアや SNS を通じて拡散される，事実とは異なる虚偽の情報のことである．その中には金銭目的や他人を困らせるために意図的に創作されるものもあれば，事実確認を怠ったことによる誤解や勘違いで偶発的に生まれたものもある．これまでも噂やデマのように，真実とは異なる情報が伝えられることはあったが，SNS の利用が浸透し，誰もが簡単に情報を共有・発信できるようになった現代では，Fake News が拡散するスピード，範囲はそれまでとは比べ物にならない．そのため Fake News は政治，経済など社会全体に影響を及ぼすようになってきており，時には大きな混乱を招くことさえ起きている．

Fake News によって生じるこうした問題に対抗するため，機械学習や自然言語処理技術を用いてニュース記事の中のねつ造や誤った情報を識別することを目的とし，Pomerleau と Rao により企画されたのが Fake News Challenge (FNC) というプロジェクト [1] である．彼らによれば，ニュースの内容が真実であるのか虚偽であるのかを評価する第一段階は「他社はどのように伝えているか」ということを知ることであるという．そしてそれを自動化すること，すなわち「スタンス検出 (Stance Detection)」こそが，あるニュースが事実であるか否かを人工知能を用いて確認する上で非常に重要な役割を果たすという．そこでスタンス検出に焦点を当て，2017 年に公開されたタスクが Fake News Challenge Stage 1 (FNC-1) [1] である．FNC-1 におけるスタンス検出のタスクは，与えられたニュース記事の見出しと本文の内容について 4 つのクラス，すなわち合致しているのか (Agree)，合致していないのか (Disagree)，同じトピックについて議論しているだけなのか (Discuss)，無関係なのか (Unrelated) ということを識別するものである．

この FNC-1 において，最も高いスコアを獲得したのは Talos Intelligence [2] というチームであった．Talos チームは畳み込みニューラルネットワーク (CNN) と Gradient-Boosted Decision Trees (GBDT) という 2 つのモデルを組み合わせてこのタスクに臨んだが，その 1 つである CNN モデルは GBDT モデルに比べ分類精度が低かった．我々はこの CNN モデルの分類精度に大きな影響を与えたものは，FNC-1 で利用されるデータセットにある偏りではないかと考えている．すなわち Talos チームが使用したモデルでは，データ数の多いクラスは正確に分類できるが，データ数の少ないクラスでは学習が不十分となり誤分類が起きる可能性がある．

そこで本研究は，Talos チームの CNN モデルに模したものを構築し，それを分類器として階層分類を行うことを目的とする．それは，階層的にクラスを分割し分類することで，データ数の不均衡を抑えることができ，さらに各クラスの特徴を捉えやすくなると考えるからである．さらに，FNC-1 のための独自の評価方法の提案も行う．FNC-1 で利用された評価方法は，データセットの偏りを考慮し重みづけを行ってはいるものの，Related (Agree, Disagree, Discuss) クラスと Unrelated クラスの 2 クラス間でしか評価の重みづけがされていないという問題を抱えている．そのため，この評価方法は，正解のスタンスが Agree であるインスタンスに対してシステムが Disagree であると予測するというリスクを全く考慮していない．そこで本研究においては，正解のスタンスとシステムが予測したスタンスの間の一致度だけでなく，不一致の度合いも定量化することができる評価方法も提案する．

2. 関連研究

2.1 FNC 類似タスク

FNC-1 のスタンス検出では，ニュース記事の見出しとその記事本文の内容との関係を識別するタスクが課題として与えられたが，類似したタスクとして行われたものに Mohammad らに

よるものと，Matsuyoshi らによるものの 2 つが挙げられる．

Mohammad らが行ったのは，SNS 上の Twitter について，Tweet した人がその内容について賛成の立場をとっているのか，それとも反対の立場なのか，あるいはそのどちらでもないのかを判断するというタスクの設定である [3]．このタスクに取り組んだほとんどのチームは n-gram や単語埋め込みベクトル，感情特性といった手法を用いてスタンス検出を行った．しかし，優れた成績を残したチームが行ったのは CNN や Recurrent Neural Network といった深層学習を用いた方法であった．

一方 Matsuyoshi らが用意したのは，NTCIR-10 RITE2 というタスクである [4]．RITE2 では，$t1$，$t2$ という 2 つのテキストがペアとして与えられ，$t1$ の内容から $t2$ の内容が真であると推論されうるかどうかを判断することが求められた．これは，与えられた 2 つのテキスト間の関係を分類するという点において FNC-1 と類似している．NTCIR-10 RITE2 に参加した Ito ら [5] は，Support Vector Machine (SVM) や Logistic Regression (LR) といった機械学習と，自らが定めたルールの両方を組み合わせたシステムを利用して，2 つのテキストの関連性を高い精度で識別した．

2.2 FNC-1 参加者らの手法

FNC-1 には世界中から 50 チームもの研究者や企業が挑戦し，事実とは異なる情報を見抜くための技術の開発を競った．例えば Riedel ら [6] は，Term Frequency (TF) と Term Frequency–Inverse Document Frequency (TF–IDF) を使って特徴を抽出し，多層パーセプトロンで分類した．彼らのこの手法は単純なものであったにもかかわらず，FNC-1 で 3 番目に高いスコアを獲得した．また Thorne ら [7] は，反駁を表す単語のカウントや TF–IDF ベクトルのコサイン類似度といった特徴を利用した多層パーセプトロンを用いた分類器や，unigram TF–IDF および bigram TF–IDF で正則化したロジスティック回帰を用いた分類器のほか，5 つの分類器にそれぞれ学習させて，それらをアンサンブルする手法を提案した．彼らの手法では，検証データにおいて分類精度の向上が見られたが，テストデータの分類は検証データよりもはるかに分類が困難であったため，全体で 11 番目のスコアを記録するにとどまった．さらに Bourgonje ら [8] は，まず n-gram マッチングで Related (Agree, Disagree, Discuss) か Unrelated かを決定し，さらに Related 中の 3 クラスについてロジステック回帰を用いて，粒度の細かい分類を行っている．FNC-1 における彼らのスタンス検出の精度はそれほど高いものではなかったが，彼らの手法は明らかにクリックを誘うような記事や噂の正確性を検出するのに有効である．

一方，Hanselowski ら [9] は FNC-1 終了後に，同タスクで優秀な成績を残した上位 3 チームのシステムを再現し，エラー分析や特徴分析を行っている．彼らは，この分析に基づき，ニュース記事の見出しと本文の語彙の重複による判別方法に，本文の意味内容の合致を識別する Long Short-Term Memory (LSTM) ネットワークを組み合わせた新たなシステム stack LSTM を提案した．そしてそのシステムは，FNC-1 上位チームに劣らない高精度な分類を実現している．

また，FNC-1 で最も優秀な成績を収めた Talos チームは，計算速度が早く実装が容易なうえに，幅広く多様なトピックを捉えることができる CNN と，見出しと本文の関係を明らかにする特徴を持ち，異なる特徴ベクトルのスケールに頑健である GBDT という 2 つのモデルを 50/50 の加重平均で組み合わせたシステムを構築した．Talos チームによれば，CNN，GBDT ともそれぞれ単独では高い精度の分類を行うことはできなかったが，アンサンブルにすることでより正確にスタンスを検出することができるという．しかし，Hanselowski らによれば，CNN モデルは GBDT モデルに比べてはるかに精度が劣っているという．すなわち CNN モデルは Agree，Disagree，Discuss，Unrelated という 4 クラスのすべてにおいて分類精度が低く，そのことがモデル全体の検出精度にマイナスの影響を与えているという．

2.3 自然言語処理における深い CNN

Conneau ら [10] は，感情分析やトピックあるいはニュース分類を行う際，CNN の畳み込み層をより深くした Very Deep Convolutional Networks (VDCNN) を提案した．VDCNN は CNN の畳み込み層を 29 層まで増やして作られたものであり，VDCNN を使うと，利用したデータセットすべてにおいて CNN より分類精度の向上が見られたという．さらに VDCNN では，ニューラルネットワークが深くなると起こる勾配消失という問題に対処するため，勾配の減衰を防ぐ残差接続 [11] が利用されている．

2.4 偏ったデータセットに対する階層分類の適用

Abdalaziz ら [12] は，多クラスデータセット内の少数のデータを正しく分類するために階層分類を提案し，不均衡なデータセットに対する階層分類の有効性を示している．彼らは，グループ内のインスタンス数が等しくなるように，もとの不均衡なデータセットを複数のグループに分割した新しいデータセットを人工的に編成し，そのデータセットを階層的なステップごとに SVM を用いて分類した．そして，その手法は従来の手法と比べても，最良の結果をもたらしたという．

3. 提 案 手 法

3.1 VDCNN モデル

本研究では，Talos チーム [2] が使用した CNN に，Conneau ら [10] の研究から着想を得て改良を加えたもの，すなわち VDCNN モデルを提案する．また，本研究でモデルを構築する際，Talos チームが GitHub 上で公開しているソースコード(注1)を参考にした．ここで，2.3 節で述べた VDCNN の利点を活用して構築したモデルの概要図を，図 1 に示す．

このモデルでは，まず FNC-1 のデータセットから得られるニュース記事の見出しと本文を別々に入力する．次に，その 2 つのテキスト入力を単語埋め込みベクトルによってベクトル化する．続いて，そのベクトル化されたテキストを Convolution Block で畳み込みを行う．この Convolution Block は 2 つの畳み込み層と，それぞれの後に続くバッチ正規化 (Batch Normalization) 層および正規化線形関数 (ReLU) から構成されてい

(注1)：https://github.com/Cisco-Talos/fnc-1

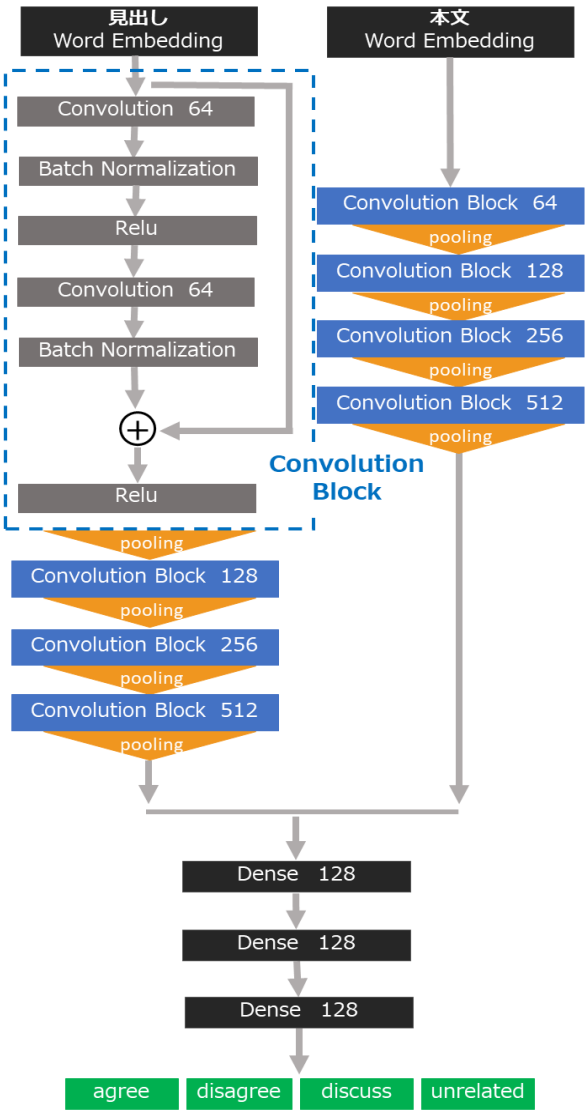

図 1　提案するモデルの概要図

る．さらに，その Convolution Block 内では残差接続が用いられている．その後 Convolution Block の出力は，Max Pooling によって重要な情報が保たれたまま情報量が圧縮され，全結合層を通過し，最終的に Agree, Disagree, Discuss, Unrelated の 4 クラスに分類される．

次に，図 1 の Convolution Block 内のネットワークについて説明する．文中で i 番目の単語に対する k 次元の単語埋め込みベクトルを $\mathbf{x}_i \in \mathbb{R}^k$ とすると，長さ n の文に対応するベクトル $\mathbf{x}_{1:n} \in \mathbb{R}^{nk}$ は以下のように表される．

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \ldots \oplus \mathbf{x}_n$$

ここで，$\oplus$ は連結演算子を意味し，$\mathbf{x}_{1:n}$ は Convolution Block への入力となる．その $\mathbf{x}_{1:n}$ は 1 層目の畳み込み層において，畳み込みウィンドウの長さが 3 で j 番目のフィルター $\mathbf{w}_j \in \mathbb{R}^{3k}$ と積和演算が行われ，バイアス b_j が付加される．その後，ReLU を用いた活性化関数 f_1 が適用されて新たな特徴 c_{ji} が生成される．これらの操作は以下の数式で表される．

$$c_{ji} = f_1(\mathbf{w}_j \cdot \mathbf{x}_{i:i+2} + b_j)$$

このフィルターが文の先頭から末尾までスライドして，出力が $\mathbf{c_j} \in \mathbb{R}^n$ という形になるようにパディングしつつ畳み込みを行うと，j 番目のフィルターによって生成される新たな特徴マップ $\mathbf{c_j}$ は以下の数式で表される．

$$\mathbf{c_j} = [c_{j1}, c_{j2}, \ldots, c_{jn}]$$

したがって，フィルターが h 個あるとき，全フィルターによって生成される特徴マップは以下の数式で表される．

$$\mathbf{C_1} = \mathbf{c_1} \oplus \mathbf{c_2} \oplus \ldots \oplus \mathbf{c_h}$$

同様にして，この $\mathbf{C_1}$ を入力した 2 層目の畳み込み層の出力は，以下の数式で表される．

$$\mathbf{C_2} = \tilde{\mathbf{c}}_1 \oplus \tilde{\mathbf{c}}_2 \oplus \ldots \oplus \tilde{\mathbf{c}}_h$$

そして，この出力に残差接続を加える．しかし，Convolution Block へ入力する $\mathbf{x}_{1:n}$ は $\mathbf{C_2}$ と次元数が異なるため，$\mathbf{x}_{1:n}$ は畳み込みウィンドウの長さが 1 で j 番目のフィルター $\mathbf{W}_j \in \mathbb{R}^k$ と積和演算が行われる．そしてバイアス B_j を付加した後は，1 層目の畳み込み層と同じ操作が行われ，h 個全てのフィルターによって新たな特徴マップが生成される．これを数式で表すと以下のようになる．

$$X_{ji} = f_1(\mathbf{W}_j \cdot \mathbf{x}_{i:i+2} + B_j)$$

$$\mathbf{X_j} = [X_{j1}, X_{j2}, \ldots, X_{jn}]$$

$$\mathbf{X_{1:h}} = \mathbf{X_1} \oplus \mathbf{X_2} \oplus \ldots \oplus \mathbf{X_h}$$

したがって，Convolution Block の最終的な出力 $H \in \mathbb{R}^{nh}$ は以下の数式で表される．

$$H = \mathbf{C_2} + \mathbf{X}_{1:h}$$

この Convolution Block は 4 種類あり，各 Convolution Block における畳み込み層が持つ出力フィルターの数は h と等しく，それぞれ 64，128，256，512 と層が深くなるにつれ増加する．

3.2　階層モデル

データセットが偏っているためにデータ数の少ないクラスを正しく分類することが困難であるという問題を解決するため，本研究では階層分類を用いた手法を提案する．2.4 節で述べたように，階層分類は偏ったデータセットに対して特に有効であり，より正確にデータ数の少ないクラスの分類ができる．ここで，本研究で提案する階層分類の概要を図 2 に示す．まず第 1 分類器を用いて，FNC-1 のデータセットにおけるインスタンスを Unrelated と Related (Agree, Disagree, Discuss) の 2 クラスに分類する．次に第 2 分類器を用いて，Related クラスに分類されたインスタンスを Agree, Disagree, Discuss の 3 クラスに分類する．

こうした第 1 分類器と第 2 分類器による階層分類を，前節で述べた VDCNN モデルと組み合わせて行い，このモデルを階層モデルと呼ぶ．なお，第 1 分類器と第 2 分類器の間でパラメータは共有されない．

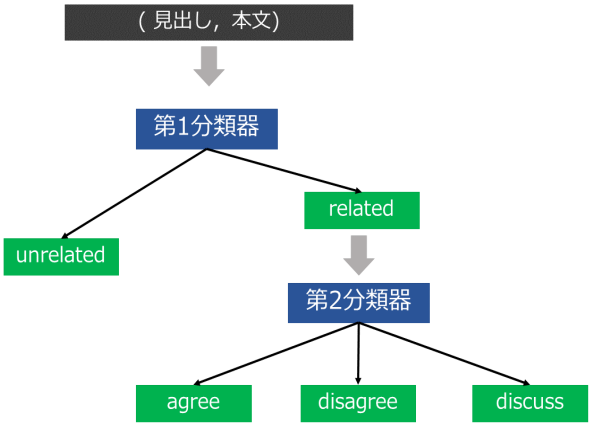

図 2　提案する階層分類の概要図

4.　データセット・評価方法

4.1　データセット

本研究でスタンス検出を行うために利用するのは，FNC-1 の主催者によって GitHub 上に公開されたデータセット (注2) である．このデータセットは，新聞記事の内容の真偽を確認するための Emergent プロジェクト (注3) からデータを抽出して作成されたものであり，Ferreira ら [13] の研究に基づいて作成されている．これら FNC-1 の主催者によって提供されたデータセットには，訓練用とテスト用の 2 種類がある．訓練用データセットには 49,972 個のインスタンスが含まれており，各インスタンスは (見出し，本文，スタンス) の組になっている．スタンスは見出しと本文の関係によって決まり，見出しと本文が合致していれば Agree，合致していなければ Disagree，ただ単に同じトピックについて議論しているだけであれば Discuss，無関係であれば Unrelated となる．訓練用データセットの分布を以下の表 1 に示す．一方，テスト用データセットには 25,413 個のインスタンスが含まれており，各インスタンスは (見出し，本文) の組となっている．FNC-1 において，訓練用データセットはテスト用データセットの各インスタンスに正しいスタンスをラベルづけするために使われ，一方，テストデータは分類器の汎化能力を確認するために使われた．そのため本研究でも，同様の目的でこれらのデータセットを利用する．

表 1　FNC-1 で用いられた訓練用データセットの分布

スタンス	Agree	Disagree	Discuss	Unrelated	Total
インスタンス (個)	3,678	840	8,909	36,545	49,972
割合 (%)	7.4	1.7	17.8	73.1	100

4.2　FNC-1 の評価方法

FNC-1 における評価は，まず見出しと本文の関係が Related (Agree, Disagree, Discuss) か Unrelated かの二値分類である．ここで正確に分類することができれば，チームのスコアは 0.25 加算される．次に Related と正しく分類されたインスタンスを Agree，Disagree，Discuss の 3 クラスのいずれかに分類する．ここでも正確に分類することができれば，チームのスコアは 0.75 加算される．

4.3　提案する評価方法

スタンス検出において，Related (Agree, Disagree, Discuss) クラスと Unrelated クラスの二値分類を誤ることと同程度に，Related クラスの 3 クラス間で誤分類をすることは深刻な問題である．しかし，FNC-1 における評価方法は，正解のスタンスとシステムの予測したスタンスが一致したときに加点していく方法であるため，Related クラス中で両者が不一致だった場合を考慮していない．例えば，実際は見出しと本文の関係が Agree であるにも関わらず，両者の関係が Disagree であると予測された場合と，両者の関係が Discuss であると予測された場合には，同じスコアがつく．しかし，Agree と Discuss という不一致よりも，Agree と Disagree という不一致のほうが深刻であると考えられる．そこで，本研究では Weighted Cohen's Kappa [14] を利用して評価する．

ここで，Weighted Cohen's Kappa の計算方法について説明する．まず正解のスタンスが第 i カテゴリ ($i=1$ のとき Agree, $i=2$ のとき Disagree, $i=3$ のとき Discuss, $i=4$ のとき Unrelated) に分類され，かつシステムの予測したスタンスが第 j カテゴリ ($j=1$ のとき Agree, $j=2$ のとき Disagree, $j=3$ のとき Discuss, $j=4$ のとき Unrelated) に分類されたときのアイテム件数を O_{ij} とする．それによって，正解のスタンスが第 i カテゴリに分類したアイテム件数を $O_{i\bullet}$，システムの予測したスタンスが第 j カテゴリに分類したアイテム件数を $O_{\bullet j}$ と表すことができる．したがって，総アイテム件数が N 個で，正解のスタンスとシステムの予測したスタンスが独立して分類が行われた場合の期待値を表す仮想データは，以下のような数式で表される．

$$C_{ij} = \frac{O_{i\bullet} O_{\bullet j}}{N}$$

さらに，各カテゴリ対の重みを W_{ij} とすると，不一致度を表す Weighted Cohen's Kappa は以下のような数式で表される．

$$\kappa = 1 - \frac{\sum_{i=1}^{4}\sum_{j=1}^{4} W_{ij}^2 O_{ij}}{\sum_{i=1}^{4}\sum_{j=1}^{4} W_{ij}^2 C_{ij}}$$

すなわちこの数式により，正解のスタンスとシステムが予測したスタンスにおける偶然以上の一致の度合いが表される．ここで，今回スタンス検出の結果を評価するにあたり設定した重みを，以下の表 2 に示す．Related と Unrelated の不一致に対しては 3，Agree と Disagree の不一致に対しては 2，Agree と Discuss の不一致あるいは Disagree と Discuss の不一致に対しては 1 の重みを設定した．また，正解のスタンスとシステムの予測したスタンスが一致したときの重みは 0 である．Weighted Cohen's Kappa を計算する際，これらの重みを 2 乗した平方重みを用いた．

(注2)：https://github.com/FakeNewsChallenge/fnc-1
(注3)：http://www.emergent.info/

表 2　各カテゴリ対の重み W_{ij}

		予測			
		Agree	Disagree	Discuss	Unrelated
正解	Agree	0	2	1	3
	Disagree	2	0	1	3
	Discuss	1	1	0	3
	Unrelated	3	3	3	0

5. 評価実験

5.1 モデルの実装

深層学習のフレームワークとして，Talos チームは Python 用の数値計算ライブラリである Theano(注4) を利用したが，Theano の開発は既に終了しているため，本研究では Keras(注5) を利用した．Keras は Python で書かれた深層学習用ライブラリであり，ニューラルネットワークの構築が Theano より容易であるという利点を持つ．さらに，本研究ではフレームワークを Theano から Keras に変更したため，利用する単語埋め込みベクトルも変更した．すなわち Talos チームは Google News のデータセットで事前学習された Word2Vec [15] を利用したが，本研究では Glove [16] で事前学習された単語埋め込みベクトル (注6) を利用した．

このような変更に伴い，本研究では Talos チームの CNN モデルを模した擬似モデルを構築し，そのモデルを Talos 擬似モデルと呼ぶことにする．そのモデルの概要図を以下の図 3 に示す．本研究で提案するモデルは Talos 擬似モデルを改良したものであるため，Talos 擬似モデルの入力と出力，全結合層は 3.1 節で述べたものと同じである．すなわち，Talos 擬似モデルと本研究で提案するモデルの違いは，畳み込み処理の方法である．Talos 擬似モデルでは，3 つ目の畳み込み層までは畳み込み層，プーリング層の順につながっており，4 つ目と 5 つ目の畳み込み層はプーリング層なしでつながっている．一方，本研究で提案するモデルでは，2 つの畳み込み層をつなげた Convolution Block の出力に残差接続を加えた後，プーリング層をつないでいる．

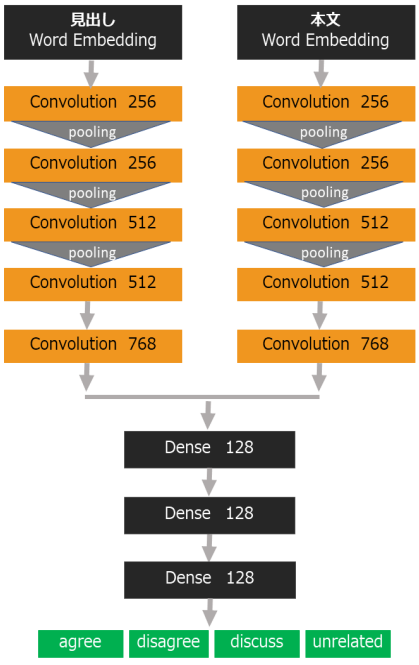

図 3　Talos 擬似モデルの概要図

(注4)：http://deeplearning.net/software/theano/
(注5)：https://keras.io/ja/
(注6)：https://nlp.stanford.edu/projects/glove/

5.2 実験結果

本節では，Talos 擬似モデルと 3.1 節で述べた VDCNN モデル，3.2 節で述べた階層モデルをそれぞれ用いた 3 つのシステムを使ってスタンス検出を行い，4.3 節で述べた評価方法である Weighted Cohen's Kappa を用いて分類精度を比較する．

まず，4.1 節で示した，FNC-1 の主催者によって提供されたデータセットの訓練用データセット (FNC-Train) を利用して，Talos 擬似モデルに学習させ，テスト用データセット (FNC-Test) によって分類精度を確認した．その結果を表 3 に示す．この表から分かるように，Talos 擬似モデルでは正しくスタンスを分類できているインスタンス数は少なく，特に Related (Agree, Disagree, Discuss) クラスのインスタンスを誤って分類することが非常に多かった．さらに，正解のスタンスと Talos 擬似モデルが予測したスタンスの一致度合いを定量化する Weighted Cohen's Kappa の値も，0.1192 と著しく低かった．しかし Talos 擬似モデルは，FNC-Train 内で訓練用とは別に切り離した検証用データセットに対しては，高い分類精度を示した．このことから，FNC-Test に対する Talos 擬似モデルの分類精度の低さは，FNC-Train と FNC-Test の性質が大きく違うことに起因するという仮説を立てた．

表 3　Talos 擬似モデルの混同行列 (FNC-Test)

		予測			
		Agree	Disagree	Discuss	Unrelated
正解	Agree	537	78	183	**1,105**
	Disagree	237	33	55	**372**
	Discuss	988	201	1,228	**2,047**
	Unrelated	3,535	770	2,298	**11,746**

この仮説を検証するために，まず FNC-Train を訓練用 (FNC-Train-Train) と，テスト用 (FNC-Train-Test) という 2 つのデータセットに分割した．ここで，FNC-Train-Train と FNC-Train-Test の総インスタンス数の割合は，FNC-Train と FNC-Test の総インスタンス数の割合と同じになるようにした．また，FNC-Train-Train における各スタンスのデータ分布は，FNC-Train における各スタンスのデータ分布と同一となり，一方，FNC-Train-Test における各スタンスのデータ分布は，FNC-Test における各スタンスのデータ分布と同一になるように設定した．そして，この FNC-Train-Train を用いて Talos 擬似モデルに学習させ，FNC-Train-Test によってテストした結果を以下の表 4 に示す．表 4 から，FNC-Train-Test における Talos 擬似モデルの分類精度が高いことは明らかであり，全インスタンスのうち 90%以上のインスタンスに対して正しく分

類できていることがわかる．また，Weighted Cohen's Kappa の値は 0.7931 と高く，FNC-Train 内では Talos 擬似モデルの学習および予測が正確に行われていることが確認できた．したがって，仮説通り Talos 擬似モデルが FNC-Test において分類精度が低いのは，FNC-Test が FNC-Train と異質なためであることが示された．

表 4　Talos 擬似モデルの混同行列 (FNC-Train-Test)

		予測			
		Agree	Disagree	Discuss	Unrelated
正解	Agree	**856**	80	40	250
	Disagree	84	**137**	11	48
	Discuss	53	10	**2,419**	488
	Unrelated	210	33	225	**11,714**

そこで，FNC-Train と FNC-Test を全て一つにまとめた後，新規に訓練用データセット (New-Train) とテスト用データセット (New-Test) にランダムに分け直した．ただし，New-Train と New-Test の総インスタンスと各スタンスのデータ分布はそれぞれ FNC-Train，FNC-Test と全く同じになるように調整した．この New-Train を用いて，Talos 擬似モデル，VDCNN モデル，階層モデルの 3 つのモデルに学習させ，New-Test によって各モデルをテストした結果を以下の表 5 に示す．この表では，上記の 3 つのモデルにおける，本研究で提案した独自の評価方法の Weighted Cohen's Kappa 値，FNC-1 における評価方法でつけられるスコア，Agree，Disagree，Discuss，Unrelated それぞれのスタンスごとの正答率が表されている．表 5 から分かるように，FNC-1 における評価方法においてだけでなく，Weighted Cohen's Kappa においても，本研究で提案した VDCNN モデルと階層モデルの分類精度は，Talos 擬似モデルよりも優れた結果を残した．さらに，最もデータ数が少なく予測が困難であると考えられる Disagree クラスについても，階層モデルの正答率は Talos 擬似モデルの正答率を大きく上回った．

表 5　新しいデータセットにおける各モデルの分類精度

モデル	Weighted Cohen's Kappa	FNC	Agree	Disagree	Discuss	Unrelated
Talos 擬似モデル	.7719	.8453	**.7047**	.5696	.8168	.9403
VDCNN モデル	.8033	.8683	.6211	.6112	**.9048**	.9448
階層モデル	**.8118**	**.8713**	.6973	**.7001**	.8647	**.9578**

5.3　信 頼 区 間

5.2 節では，Talos 擬似モデルと VDCNN モデル，階層モデルの 3 モデルに対する Weighted Cohen's Kappa 値を算出した．そこで本節では，上記の 3 モデルにおける Weighted Cohen's Kappa の 95%信頼区間 (注7) を求め，統計的に本質的な差があるかどうか議論する．

まず，κ 統計量の $(1-\alpha)100\%$信頼区間の計算方法について説明する．上側確率が $\alpha/2$ に対応する正規分布のパーセント点を $Z_{\alpha/2}$ とすると，κ 統計量の $(1-\alpha)100\%$信頼区間は以下の式で表される．

$$\kappa \pm Z_{\alpha/2} \times \sigma_k$$

ここで，4.3 節で述べた変数 N，O_{ij}，C_{ij}，W_{ij} を利用して，

$$Q_o = \frac{\sum_{i=1}^{4}\sum_{j=1}^{4} W_{ij}^2 O_{ij}}{N}, Q_c = \frac{\sum_{i=1}^{4}\sum_{j=1}^{4} W_{ij}^2 C_{ij}}{N}$$

$$R_o = \frac{\sum_{i=1}^{4}\sum_{j=1}^{4} (W_{ij}^2)^2 O_{ij}}{N}$$

とすると，標準偏差 σ_k は，

$$\sigma_k = \sqrt{\frac{R_o - Q_o^2}{NQ_c}}$$

と表される．

以上の式を用いて算出した，Weighted Cohen's Kappa の 95%信頼区間は，Talos 擬似モデルで $[0.7634, 0.7805]$，VDCNN モデルで $[0.7954, 0.8113]$，階層モデルで $[0.8039, 0.8198]$ となった．これを視覚化すると以下の図 4 のようになる．図 4 から分かる通り，Talos 擬似モデルと VDCNN モデル間，Talos 擬似モデルと階層モデル間には本質的な差がある．一方，VDCNN モデルと階層モデル間には本質的な差がほとんど見られない．しかし，統計的に差がないように見える分類器であっても，どのようなときに正しく分類できて，どのようなときに間違った分類をしてしまうのかという傾向を知ることで，実用性の違いを見ることができる．そこで，次節では具体的なインスタンス (見出し，本文，モデルが分類したスタンス) を挙げながら，実験結果に対しての詳しい考察を行う．

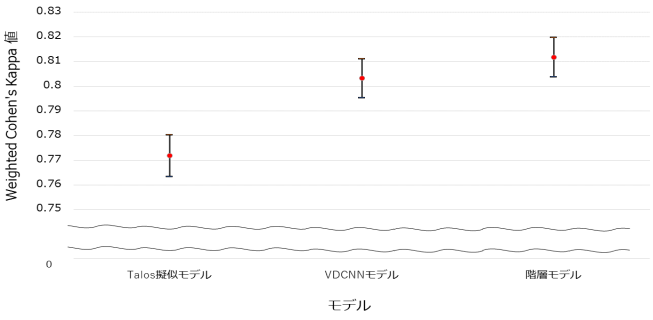

図 4　各モデルにおける Weighted Cohen's Kappa 値とその 95%信頼区間

6.　考　　察

6.1　Talos 擬似モデルと VDCNN モデルの比較

表 5 に示すように，VDCNN モデルでは Talos 擬似モデルと比べ，Disagree クラスと Discuss クラスの正答率が向上した．それは，Talos 擬似モデルでは局所的な特徴に影響されて誤ったスタンスを予測してしまう問題が生じていたが，文中の特徴をより抽象化して捉える VDCNN モデルではこの問題にうまく対処できたからであると考えられる．すなわち，CNN の層

(注7)：http://aoki2.si.gunma-u.ac.jp/lecture/Kappa/

が深くなったことで，文全体としてのセマンティックな特徴を捉えることができるようになったのである．

ここで，Talos 擬似モデルはスタンスを誤分類したが，VDCNN モデルは正しくスタンスを分類できた例を挙げる．見出しに “Rumor: Seth Rogen to Appear as Woz in Sony’s Steve Jobs Film” とあり，本文にも “Seth Rogen” や “Steve Jobs”，“appear” といった単語が含まれ,「Seth Rogen が Sony の Steve Jobs Film に出演する」ということを推測させるような内容が含まれている記事を，Talos 擬似モデルは Agree と判断した．しかし記事本文には，“will reportedly appear” や “highly anticipated” のように未来を予測させる言葉が使われており，明確に「Seth Rogen が Sony’s Steve Jobs Film に出演する」と言及している箇所は見られない．そのため正解は Agree ではなく，Discuss であり，Talos 擬似モデルは分類を誤っている．一方 VDCNN モデルは，見出しと本文の関係を Discuss と正しく判定した．このように VDCNN モデルが正しく分類できたのは，記事本文に含まれたセマンティックな特徴を捉えることができたためだと考えられる．

6.2 VDCNN モデルと階層モデルの比較

5.3 節で示したように，本研究で提案する評価方法において，VDCNN モデルと階層モデルの間に本質的な差はほとんどない．しかし，Talos 擬似モデル，VDCNN モデル，階層モデルの 3 モデルの中で，階層モデルは Disagree クラスと Unrelated クラスの正答率が最も高かった．これは，第 1 分類器と第 2 分類器に分けて階層的に分類することで，データの偏りが抑えられ，各クラスが持つ特徴をより捉えやすくなったためだと考えられる．特に，データ数が少なく，文の構成や文中に出現する単語が Agree クラスや Discuss クラスと類似しているため，スタンスの明確な判断が難しい Disagree クラスに対して，第 2 分類器は他クラスとの違いを識別することができた．

ここで，VDCNN モデルはスタンスを誤分類したが，階層モデルは正しくスタンスを分類できた例を挙げる．見出しは “New iOS 8 bug can delete all of your iCloud documents” であるが，本文には “Elected officials and activists in New York are reacting positively to the city’s new policy...” とあり，両者が無関係であることが一目瞭然であるインスタンスに対して，VDCNN モデルは両者の関係が Agree であると誤認識した．これは，見出し文中の “bug” や “delete” といった表現と，本文中の “arrest” や “crime” といった表現から，ネガティブな内容であるという点で合致していると判断したためだと考えられる．一方，階層モデルは，第 1 分類器において，Agree クラスだけでなく，Disagree クラスや Discuss クラスといった Related クラス全体に見られる特徴を考慮して学習したため，この見出しと本文は Related クラスのインスタンスとは異質である，つまり Unrelated クラスであると正しく分類することができた．このような場合において，前述したような特性を持ち，高い分類精度を有する階層モデルは，VDCNN モデルと比べてより実用的である．

6.3 失敗分析

5.2 節で比較した 3 つのモデルに対して失敗分析を行った結果，以下のような場合にこれらのモデルがスタンスの分類を誤っていることがわかった．

(1) 記事本文に画像や動画の内容の説明がない

見出しは本文に挿入された動画の説明であり，記事本文にはその動画の内容についての記述がないという場合，正確な分類に失敗する確率が高くなる．記事本文に画像や動画の内容についての説明がなければ，動画の内容について述べる見出しと本文の関係を判断することは当然難しくなる．ニュース記事の本文に画像や動画が挿入されていることは珍しくないが，見出しが画像や動画の説明であり，記事本文には画像や動画の具体的な内容が記載されていない場合，見出しと記事本文の間に関連性があると判断するための情報が充分に得られず，誤分類に至る傾向がある．

(2) 文中に未知語が含まれている

見出しおよび記事本文に未知語，すなわち事前学習された単語埋め込みベクトルの辞書に存在しない語が含まれているために，スタンスの分類に失敗してしまうケースがある．通常，未知語は一般的ではない新語や造語であるため，見出しや記事本文全体の中で果たす役割は小さい．しかし，ある単語が見出しと記事本文の両方で使われ，さらにそれが未知語と判断されるとき，見出しと記事本文の間には共通の話題がないと判断される可能性がある．

(3) 記事本文が短い

記事本文が極端に短いために，モデルがスタンスを誤分類してしまう場合がある．記事本文が短いと，モデルはスタンスを分類するのに充分な情報を得ることができない．そのような状況下で，見出しと本文に関連性があると判断することは非常に難しい．正確なスタンス分類には，分類器の精度だけでなく，与えられる情報の量が大きく関係することがわかる．

(4) 記事本文の内容が複数

見出しと記事本文の内容が同じ話題を扱っているにも関わらず，Agree か Disagree かを正確に判断できないケースがある．例えば，記事本文の前半と後半で，述べている内容が異なっているようなインスタンスの場合である．今回実験を行った 3 つのモデルは，このような複雑な文構成の場合，その文脈・要点を捉えることができず，見出しと本文の関係が Agree なのか Disagree なのかの判断に失敗した．

(5) 正解が間違っている

ごく稀であるが，見出しと記事本文の関係に対するスタンスの正解が間違っている可能性が存在する．例えば，正解のスタンスは Disagree とあるが，記事本文は詳細について述べていないため，はっきり Disagree だと決定づけることができないというようなインスタンスがある．このとき，Disagree ではなく，Discuss と判断したモデルが正しい可能性もある．

7. 結論と今後の課題

本研究では，ニュース記事の見出しと本文の関係を識別する Fake News Challenge Stage 1 (FNC-1) というタスクに対し，畳み込みニューラルネットワーク (CNN) をベースとし，畳み込み層をより深くして残差接続を用いた VDCNN モデルに階

層分類を組み合わせた階層モデルを用いて，スタンス検出を行った．さらに，正解のスタンスとシステムが予測したスタンスの間の一致度だけでなく，不一致の度合いも定量化することができる Weighted Cohen's Kappa に基づく独自の評価方法も提案した．そして評価実験の結果，本研究で提案した階層モデルは，FNC-1 で使用された評価方法および本研究で提案された評価方法の両方において，分類精度の向上が見られた．そして，この分類精度の向上について統計的な面から議論するだけでなく，具体的なインスタンスを挙げた精細な考察を行うことで，実用的な差があることも確認できた．さらにデータセットを分析することで，FNC-1 の主催者によって提供された訓練用データセットとテスト用データセットは性質が大きく異なり，そのことが分類精度に影響しているということも確かめることができた．

今後は，スタンス検出のさらなる精度向上に向け，文中に現れる未知語への対応や，テキストだけでなく画像・動画認識による情報も考慮した総合的な判断を行うシステムの開発が必要である．また，本研究で提案したモデルと，他の優れた手法を組み合わせたシステムの開発も，将来の課題として挙げられる．

文　献

[1] Dean Pomerleau and Delip Rao. Fake News Challenge, 2017. http://www.fakenewschallenge.org/.

[2] Sean Baird, Doug Sibley, and Yuxi Pan. Talos Targets Disinformation with Fake News Challenge Victory, 2017. https://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html.

[3] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. SemEval-2016 Task 6: Detecting Stance in Tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 31–41. Association for Computational Linguistics, 2016.

[4] Suguru Matsuyoshi, Yusuke Miyao, Tomohide Shibata, Chuan-Jie Lin, Cheng-Wei Shih, Yotaro Watanabe, and Teruko Mitamura. Overview of the NTCIR-11 Recognizing Inference in TExt and Validation (RITE-VAL) Task. In *NTCIR*, 2014.

[5] Daiki Ito, Masahiro Tanaka, Hayato Yamana, et al. WSD Team's Approaches for Textual Entailment Recognition at the NTCIR10 (RITE2). In *NTCIR*, 2013.

[6] Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. A Simple but Tough-to-beat Baseline for the Fake News Challenge Stance Detection Task. *arXiv preprint arXiv:1707.03264*, 2017.

[7] James Thorne, Mingjie Chen, Giorgos Myrianthous, Jiashu Pu, Xiaoxuan Wang, and Andreas Vlachos. Fake News Detection using Stacked Ensemble of Classifiers. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pp. 80–83, 2017.

[8] Peter Bourgonje, Julian Moreno Schneider, and Georg Rehm. From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pp. 84–89, 2017.

[9] Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. A Retrospective Analysis of the Fake News Challenge Stance-Detection Task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 1859–1874. Association for Computational Linguistics, 2018.

[10] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very Deep Convolutional Networks for Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 1107–1116. Association for Computational Linguistics, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[12] Hanaa S Abdalaziz and Fakhreldeen A Saeed. New Hierarchical Model for Multiclass Imbalanced Classification. *Journal of Theoretical & Applied Information Technology*, Vol. 95, No. 16, 2017.

[13] 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*, pp. 1163–1168. Association for Computational Linguistics, 2016.

[14] 酒井哲也. 情報アクセス評価方法論　検索エンジンの進歩のために, pp. 198–205. コロナ社, 2015.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[16] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. Association for Computational Linguistics, 2014.