Vol.2015-NL-223 No.3
2015/9/27

畳み込みニューラルネットワークを用いた複単語表現の解析

進藤 裕之[1,a)]　松本 裕治[1,b)]

概要：本稿では，畳み込みニューラルネットワークを用いて文字，単語，複単語表現の特徴量をデータから自動的に学習し，文に含まれる複単語表現の同定および品詞タグ付けを行う手法を提案する．提案手法は，従来の固有表現認識やチャンキングの手法と異なり，句動詞のように単語が文中で連続していない場合にも頑健に複単語表現を同定することができる．評価実験の結果，提案手法は規則ベースの手法や系列ラベリングに基づく既存手法の性能を上回ることを確認した．

1. はじめに

複単語表現（multiword expression）とは，複数の単語で構成され，それ全体で特有の意味解釈を持つ表現の総称である [9]．例えば，英語の“ according to ”や“ a number of ”などの表現は，全体として一つの意味を成し，統語的には機能語の役割を果たす．複単語表現を正しく認識することは，自然言語の意味理解に必要不可欠であり，実際に，構文解析，機械翻訳，文生成などの自然言語処理タスクの性能向上に大きく寄与することが示されている [1], [6], [7]．

これまでの複単語表現に関する研究では，固有表現，複合名詞，句動詞，軽動詞（light-verb construction）などの特定の複単語表現を対象にして，それぞれ個別に言語資源の構築や解析手法の提案が行われてきた [5], [8], [13], [14]．そのため，様々な種類の複単語表現を網羅的に注釈したコーパスはあまり整備されておらず，標準のベンチマークデータで解析手法の良し悪しを比較することが難しかった．その結果，英語やその他の西洋語のように単語をスペースで区切る言語の構文解析や意味解析では，依然として単語を基本単位とすることを前提としており，複単語表現に十分な注意が払われていないのが現状である．

このような状況を打開するために，近年，包括的な複単語表現のリソース構築が精力的に行われている [11], [12], [15]．これらのリソースは，Penn Treebank 形式の構文木コーパス上に複単語表現の情報を付与しているため，同じテキストデータを用いて複単語表現の解析や構文解析の評価を行うことができる．したがって，今後は，従来の品詞タグ付けや構文解析と，複単語表現の解析とを統合することによって，より高度な意味解析の実現に寄与することが期待できる．

そこで本研究では，複単語表現の同定と品詞タグ付けを同時に行うタスクに焦点を当てる．具体的には，入力文を単語または複単語表現に分割し，それぞれのトークンに品詞タグを付与する．これは，日本語では形態素解析と呼ばれるタスクと類似しているが，英語の場合は，複単語表現を構成する単語が常に連続して文中に出現するとは限らないという点が異なる．例えば，英語の句動詞“ bring up ”は，“ bring someone up ”というように，単語や句が複単語表現の間に入り込むことがあるため，従来の固有表現認識やチャンキング等で用いられてきた系列ラベリングの手法をそのまま適用することはできない．

このような非連続なパターンに対しても頑健に複単語表現の同定を行うために，我々は，畳み込みニューラルネットワークを用いて単語や複単語表現の特徴量を抽出する手法を提案する．我々の提案するニューラルネットワークは，「文字の組み合わせが単語を構成し，単語の組み合わせが複単語表現を構成する」という言語の持つ階層性を自然に統計モデル化したものであり，連続・非連続のパターンを問わず，複単語表現の解析に有用な特徴量をデータから自動的に学習することができる．本研究では特に，Shigeto ら [12] が英語の Penn Treebank コーパスに注釈を付与した複合機能語のデータと，駒井ら [15] が英語の OntoNotes コーパスに注釈を付与した句動詞のデータを併合し，それらの複単語表現の同定および品詞タグ付けを行うタスクに取り組む．評価実験の結果，提案手法は規則ベースの手法や，近年提案された非連続パターンを扱える系列ラベリングの手法と比較して，性能が上回ることを確認した．

[1] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
[a)] shindo@is.naist.jp
[b)] matsu@is.naist.jp

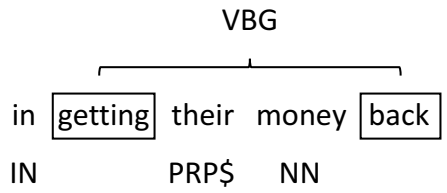

図 1 入力 “ in getting their money back ” に対する複単語表現の同定と品詞タグ付けの例.

2. 関連研究

Shigeto ら [12] は，英語の Penn Treebank コーパス上に機能語（主に副詞，接続詞，限定詞，前置詞）となる複単語表現の注釈を付与した．また，条件付き確率場（CRF）を用いて複単語表現の同定と品詞タグ付けを同時に行う手法を提案している．ただし，彼らの解析手法は，複単語表現を構成する単語が文中で連続して出現することを仮定しており，句動詞のような非連続のパターンへそのまま適用することができない．Constant ら [3] も同様に，フランス語の複単語表現を対象として，条件付き確率場による連続パターンの同定手法を提案している．

Schneider ら [10] は，英語の Web Treebank コーパスの一部（3812 文）に，連続，非連続パターンを問わず，網羅的に複単語表現の注釈を付与した．また，固有表現認識やチャンキングで用いられてきた BIO タグセットを拡張し，限定的ではあるが，系列ラベリングの手法によって非連続な複単語表現を同定する手法を提案している．彼らの手法は，人手で設計された素性テンプレートを用いて単語や文脈情報の特徴量を抽出し，文中の各単語に対して順番に拡張 BIO タグセットのいずれかを付与していく．このような系列ラベリングの手法は，学習データに出現しない複単語表現も同定することができるため，固有表現や複合名詞のような生産性の高い複単語表現を同定する場合には特に有効であると考えられる．一方，我々の手法は，人手による素性テンプレートが不要で，複単語表現の同定や品詞タグ付けに有用な特徴量をデータから自動的に学習する．また，学習データから構築した複単語表現辞書のエントリと一致した単語についてのみ，複単語表現かどうかを判別する．このような手法は，機能表現や動詞句などの生産性があまり高くない複単語表現の解析に適していると考えられる．

3. 畳み込みニューラルネットワークを用いた複単語表現解析

3.1 問題設定

本研究で取り組むタスクは，入力文の単語列に含まれる複単語表現を同定し，各トークンに品詞を付与することである．これを，「複単語表現解析」と呼ぶことにする．図 1 に，入力文の一部 “ in getting their money back ” に対する複単語表現解析の結果を例示する．図 1 では，“ getting back ” が一つの複単語表現で，品詞が VBG（動名詞）であることを表している．また，複単語表現以外のトークン（単語）に対しても，IN（前置詞），PRP$（所有代名詞）などの品詞を付与する．文中に “ getting ... back ” が出現したとき，それが必ず複単語表現の用法として用いられているとは限らない．そのため，文中の複単語表現の候補に対して，それが複単語表現としての用法なのかを判別する必要がある．また，単語の品詞タグセットと複単語表現の品詞タグセットは必ずしも同じである必要はないが，今回我々が実験で用いるデータでは，どちらも英語の OntoNotes コーパスで用いられる共通の品詞タグセット（49 種類）となっている．

複単語表現の同定や品詞タグ付けの精度は，入力文からどのような特徴量（素性）を抽出するかということに大きく依存する．単語の品詞を決定する際には，対象となる単語自身とそれに含まれる文字，周囲の単語や文字，またはそれらの組み合わせなどが特徴量となりうる．複単語表現の場合は，それらの特徴量に加えて，複単語表現自身を構成する単語，品詞，さらには単語の組み合わせなども考慮しなければならない．これら特徴量の組み合わせは膨大な数となり，その中から有効な特徴量を実験的に探すのは多大な労力がかかる．

そこで我々は，ニューラルネットワークを用いて，文字，単語，複単語の各レベルで有効な特徴量を自動的に学習することのできるモデルを提案する．図 3 に提案モデルの模式図を示す．提案モデルは，畳み込みニューラルネットワークと呼ばれるネットワーク構造を文字から単語，単語から複単語へと階層的に積み重ねることによって，複単語表現の解析に有効な特徴量をデータから自動的に学習する．最終的に得られた特徴ベクトルは線形分類器に入力され，複単語表現の同定や，各トークンに対する品詞タグの決定が行われる．

ニューラルネットワークを用いて単語や文字の特徴量を学習し，単語の品詞タグ付けを行う手法は既にいくつか提案されており [2], [4]，我々の研究は，彼らのモデルを単語から複単語表現へ，連続から非連続なパターンまで扱えるように拡張したものとみなすことができる．

我々が実験で用いるデータは，Shigeto ら [12] が英語の Penn Treebank コーパスに注釈を付与した機能表現（主に副詞，接続詞，限定詞，前置詞となる複単語表現）と，駒井ら [15] が英語の OntoNotes コーパスに注釈を付与した句動詞のデータを併合したものである．どちらのコーパスも Wall Street Journal 部を含んでおり，Shigeto ら [12] が注釈を付与した Penn Treebank と同じ文が OntoNotes にも含まれている場合に，注釈情報を OntoNotes の対応する文へ移行した．Shigeto ら [12] のデータは，連続したパター

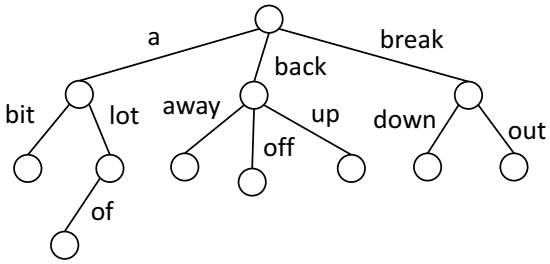

図 2 複単語表現を格納したトライ木構造の例.

ンの複単語表現のみを対象としており，例えば，“ as well as ”や“ according to ”などの機能表現が含まれている．一方，駒井ら [15] のデータは，“ bring up ”や“ look for ”などの句動詞を対象としているため，非連続パターンも含まれている．

3.2 複単語表現解析の流れ

畳み込みニューラルネットワークを用いた複単語表現解析の流れは以下の通りである．

(1) 入力文の単語列から，複単語表現の候補を検索する．
(2) 複単語表現の候補となる単語列に対して，図 3 の畳み込みニューラルネットワークを用いて特徴ベクトルを計算し，候補の単語列が複単語表現であるかどうかを判別する．複単語表現である場合，同じ特徴ベクトルを用いて最適な品詞を付与する．
(3) 複単語表現ではない単語も同様に，図 3 の畳み込みニューラルネットワークの一部（文字 - 単語レイヤーまで）を用いて特徴ベクトルを計算し，単語ごとに独立に品詞を決定する．

以下，それぞれの処理について詳細を説明する．

3.3 複単語表現の検索

入力文に含まれる複単語表現の候補を素早く検索するために，単語をエッジとするトライ木構造を用いる．図 2 にトライ木構造の例を示す．まず，あらかじめ学習データに含まれる複単語表現をトライ木構造へ格納しておく．そして，入力文の単語を左から順番に見ていき，トライ木構造に含まれるかどうかを検索する．もし含まれていれば，トライ木のノードをさらに深くたどっていくことによって，文中の単語列が複単語表現の候補になりうるかどうかを判定できる．トライ木構造は，前方一致検索が可能なので，非連続なパターンや 3 単語以上の複単語表現を効率良く検索できるという利点がある．

3.4 畳み込みニューラルネットワークによる特徴ベクトルの学習

図 3 に示されているように，提案モデルは，文字列から単語の特徴ベクトルを計算する畳み込みニューラルネットワークと，単語列から複単語表現の特徴ベクトルを計算する畳み込みニューラルネットワークで構成されている．

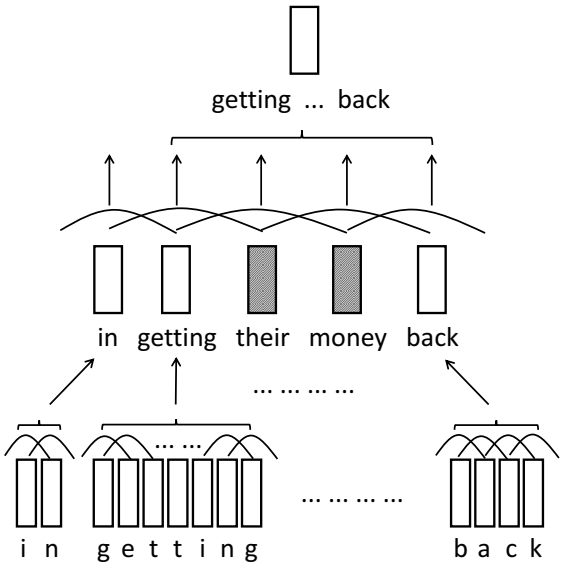

図 3 複単語表現解析のための畳み込みニューラルネットワーク構造の概要．縦長の長方形は，ニューラルネットワークの入出力となる特徴ベクトルを模式的に表したものである．

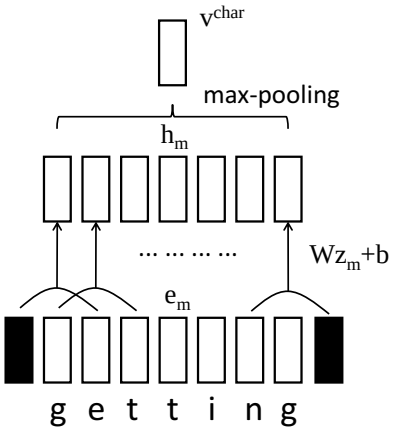

図 4 文字列の特徴量を抽出する畳み込みニューラルネットワーク構造．縦長の長方形はベクトルを模式的に表したものである．黒塗りの長方形は，パディングを表すベクトルである．図は，窓幅 $k = 3$ の場合である．

3.4.1 文字列の特徴ベクトル

未知語に対しても頑健に品詞タグ付けを行うためには，文字レベルでの特徴量が必要となる．そのために，文字列の特徴量を計算する畳み込みニューラルネットワークを構築し，単語の特徴ベクトルの一部とする．図 4 にネットワーク構造の概要を示す．図 4 は，図 3 中で，文字から単語の特徴ベクトルを計算する部分をより詳細に記述したものである．図 4 に示すように，単語を構成する文字はそれぞれベクトル表現に変換され，最終的に文字列の情報を統合した特徴ベクトル v^{char} を得る．具体的な計算手順は以下の通りである．

まず，テキストに含まれるあらゆる文字 c について，

d_1^{char} 次元のベクトル $e_c \in \mathbb{R}^{d_1^{char}}$ を用意する．e_c は c と一対一に対応しており，埋め込みベクトル（embedding）とも呼ばれる．このとき，単語 w を構成する文字列を $\{c_1, c_2, \ldots c_M\}$ とすると，w に対応する文字埋め込みベクトル列は，$\{e_{c_1}, e_{c_2}, \ldots, e_{c_M}\}$ となる．

畳み込みニューラルネットワークは，畳み込み層と pooling 層と呼ばれる二つの関数を順番に適用することによって，文字埋め込みベクトル列を特徴ベクトルへと変換する．畳み込み層では，まず m 番目の文字 c_m に対して，c_m を中心とした文字 n-gram の埋め込みベクトルを連結し，z_m とする．

$$z_m = \mathrm{concat}\left(e_{m-\lfloor k^{char}/2 \rfloor}, \ldots, e_{m+\lfloor k^{char}/2 \rfloor}\right)$$

ただし，concat は，引数となる全てのベクトルを連結した新たなベクトルを返す関数である．また，k^{char} は窓幅，$\lfloor \cdot \rfloor$ は床関数である．例えば，$k^{char} = 5$ の場合，m を中心に左右 2 文字ずつを含めた文字埋め込みベクトルを連結したものが z_m である．先頭や最後の文字では，左または右の文字が存在しないので，パディングと呼ばれる特別なベクトルを代わりに使う（図 4 の黒い長方形）．

次に，z_m に対してアフィン変換を行い，d_2^{char} 次元のベクトル $h_m \in \mathbb{R}^{d_2^{char}}$ を得る．

$$h_m = W^{char} z_m + b^{char}$$

ただし，$W^{char} \in \mathbb{R}^{d_2^{char} \times d_1^{char} k^{char}}$ は重み行列，$b^{char} \in \mathbb{R}^{d_2^{char}}$ はバイアスベクトルである．

その後，pooling 層では，文字ごとに計算したベクトル h_m を一つのベクトルへまとめる処理を行う．様々な種類の pooling 関数が存在するが，我々は，代表的な pooling 関数の一つである max-pooling を用いる．max-pooling 関数は，同じ次元数の複数のベクトルに対して，各次元で最も値の大きな値のみを出力する．具体的には，max-pooling 関数の出力を v^{char} とすると，$v^{char} \in \mathbb{R}^{d_2^{char}}$ は以下のように計算される．

$$\left[v^{char}\right]_j = \max_{1<m<M} \left[h_m\right]_j$$

ただし，$\left[h_m\right]_j$ は，h_m の j 次元目の値を指す．

直観的には，畳み込みニューラルネットワークは単語に含まれる全ての文字 n-gram を取り出して線形変換を行い，max-pooling 関数によって有効な特徴量を取り出していると考えられる．単語の品詞タグ付けの場合には，畳み込みニューラルネットワークの学習を行うことによって，単語の接頭辞や接尾辞の文字情報が max-pooling 関数で抽出されることが期待できる．

3.4.2 単語の特徴ベクトル

単語の特徴ベクトルは，前述のニューラルネットワークによって文字列から計算された特徴ベクトルと，単語そのものの埋め込みベクトル e^{word} を連結したものとする．すなわち，d^{word} 次元の単語の特徴ベクトル $v^{word} \in \mathbb{R}^{d^{word}}$ を以下のように計算する．

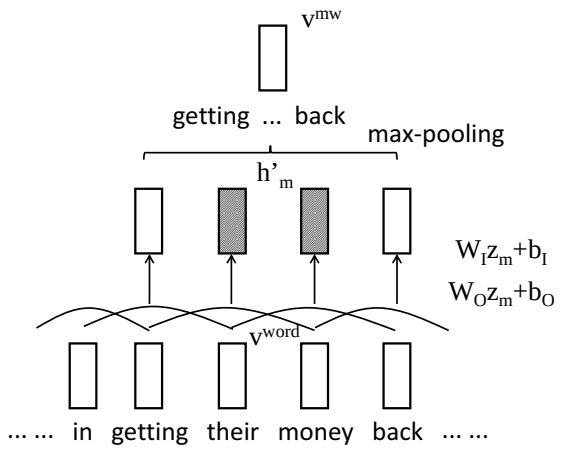

図 5 単語列から複単語表現の特徴量を抽出する畳み込みニューラルネットワーク構造．縦長の長方形はベクトルを模式的に表したものである．黒く塗られた長方形は，複単語表現に含まれない内部の単語ベクトルを表す．

$$v^{word} = \mathrm{concat}\left(e^{word}, v^{char}\right)$$

3.4.3 複単語表現の特徴ベクトル

次に，単語の特徴ベクトルから，複単語表現の特徴ベクトルを計算する畳み込みニューラルネットワークを構築する．図 5 にネットワーク構造の概要を示す．図 5 は，図 3 中で，単語から複単語表現の特徴ベクトルを計算する部分をより詳細に記述したものである．

まず，複単語表現の候補となる単語列を $\{w_1, \ldots, w_M\}$ とする．このとき，$\{w_1, \ldots, w_M\}$ に対応する単語の特徴ベクトルは $\left\{v_1^{word}, \ldots, v_M^{word}\right\}$ となる．複単語表現を構成する単語が文中で非連続の場合には，複単語表現の先頭の単語を w_1，最後の単語を w_M と定義し，先頭から最後の単語までの連続した単語列を指すこととする．例えば，図 5 で示す “ in getting their money back ” の場合には，“ getting ... back ”が複単語表現の候補となるので，“ getting ”が w_1，“ back ” が w_4 となり，$M = 4$ である．

まず，畳み込み層では，先ほど同様に，w_m を中心とした単語 n-gram の特徴ベクトルを連結して z'_m とする．

$$z'_m = \mathrm{concat}\left(v^{word}_{m-\lfloor k^{word}/2 \rfloor}, \ldots, v^{word}_{m+\lfloor k^{word}/2 \rfloor}\right)$$

ただし，k^{word} は窓幅である．

次に，z'_m に対してアフィン変換を行い，d^{mwe} 次元のベクトル $h'_m \in \mathbb{R}^{d^{mwe}}$ を計算する．ただし，非連続パターンの複単語表現に対処するために，$\{w_1, \ldots, w_M\}$ の中で複単語表現の構成要素となる単語と，それ以外の単語とを区

別し，それぞれ異なるパラメータでアフィン変換を行う．すなわち，

$$h'_m = \begin{cases} W_I^{word} z'_m + b_I^{word} & \text{if } m \in P \\ W_O^{word} z'_m + b_O^{word} & \text{otherwise} \end{cases}$$

ここで，W_I^{word}, b_I^{word} は複単語表現の構成要素となる単語に適用するパラメータ，W_O^{word}, b_O^{word} はそれ以外の内部単語に適用するパラメータである．P は複単語表現の構成要素となる単語のインデックス m の集合である．

その後，pooling 層では，単語ごとに計算したベクトル h'_m を一つのベクトルへまとめる処理を行う．文字の畳み込みニューラルネットワークと同様に，max-pooling 関数を用いる．

$$[v^{mwe}]_j = \max_{1<m<M} [h'_m]_j$$

以上の計算によって，複単語表現の特徴ベクトルを計算することができる．

複単語表現の構成要素とはならない単一の単語についても，1 単語の複単語表現とみなすことによって，上記と同じ計算手順によって特徴ベクトルを計算する．ただし，1 単語の場合は max-pooling 関数の入出力が同じベクトルになってしまうため，代わりに活性化関数 tanh を用いる．

3.5 複単語表現の同定と品詞タグ付け

上記の畳み込みニューラルネットワークによって計算された特徴ベクトルを用いて，複単語表現の同定と，各トークンに対する品詞タグ付けを行う．

まず，文中に含まれる複単語表現の候補は，特徴ベクトル v^{mwe} を入力として，線形分類によって複単語表現であるかを判別する．複単語表現であると判別された場合には，同じ特徴ベクトルを使って線形分類で品詞を決定する．実際には，複単語表現が複数の品詞を取ることは稀であるため，複単語表現であると判別されれば，その品詞はほぼ一意に定まる．

入力文に含まれる全ての複単語表現を同定した後，残りの複単語表現ではない単一の単語について品詞を決定する．これも先ほどと同様に，単語の特徴ベクトル v^{mwe} を線形分類器に入力し，最適な品詞タグを決定する．ただし，複単語表現の線形分類器と，単語の品詞を決定する線形分類器は別々に用意する．

提案モデルの学習は，確率的勾配法によって行う．正解の複単語表現の範囲とその品詞，それ以外の各単語の品詞を訓練データとして与えて，モデルの対数尤度が最大になるようにパラメータを更新していく．パラメータの更新は，一文ごとにまとめて行った（ミニバッチ学習）．

ハイパーパラメータ	値
d_1^{char}	10
d_2^{char}	50
d^{word}	150
d^{mwe}	300
k^{char}, k^{word}	5

表 1 実験で用いた提案モデルのハイパーパラメータ一覧．

	適合率	再現率	F 値
規則ベース	95.9	96.7	96.3
拡張 BIO 系列ラベリング [10]	96.8	96.7	96.7
提案手法	**97.3**	**97.3**	**97.3**

表 2 全トークンに対する複単語表現解析の性能

	適合率	再現率	F 値
規則ベース	76.1	92.3	83.4
拡張 BIO 系列ラベリング [10]	93.3	90.0	91.6
提案手法	**92.2**	**93.5**	**92.8**

表 3 複単語表現（2 単語以上で構成されるトークン）のみの解析性能

4. 実験

4.1 実験設定

提案手法の有効性を検証するため，複単語表現解析の実験を行った．実験で用いるデータは，3.1 節で述べたように，英語の OntoNotes コーパスの Wall Street Journal 部に対して機能表現および句動詞の注釈が付与されたものである．各注釈は，コーパスの文番号，複単語表現の位置，複単語表現の品詞，を含む．元々の OntoNotes コーパスには，各単語に品詞タグの情報が付与されているため，この単語品詞タグと，複単語表現の注釈情報を組み合わせて学習データおよびテストデータを構築した．学習データは Wall Street Journal のセクション 02-21，テストデータはセクション 23 とした．学習データに含まれる複単語表現の数は全部で 10798 事例あり，そのうち 434 事例が非連続パターンであった．

実験では，規則に基づく複単語表現解析手法と，Schneider ら [10] によって提案された系列ラベリングの手法と提案手法とを比較した．規則ベースの解析手法として，複単語表現辞書にマッチした候補に対して，それらが文中で連続して出現している場合には複単語表現である，そうでない場合には複単語表現ではないという単純な規則を設けた．単語の品詞は，後述する Stanford POS Tagger を用いて付与し，複単語表現の品詞は，データ中で最も頻出した品詞を付与した．

Schneider ら [10] の手法は，固有表現やチャンキングで用いられてきた BIO タギングによる複単語表現の同定手

法を，非連続なパターンへ拡張したものである．彼らの手法では，{B, I, O, b, i, o} の 6 種類のタグを用意して，各単語にいずれかのタグを付与する．{B, I, O} はそれぞれ，複単語表現の開始単語，内部または終了単語，外側の単語，を表す．{b, i, o} は {B, I, O} と同じ意味であるが，ある複単語表現に別の複単語表現が埋め込まれている場合に，内部の複単語表現の開始位置や終了位置を表すために用いられる．例えば，入力が“ in getting their money back ”のとき，正解のタグ系列は“ O B o o I ”となる．彼らの手法では，入力単語に品詞が付与されていることを前提としているため，我々はあらかじめ Stanford POS Tagger [*1] を用いて，10-fold のジャックナイフ法で学習データの品詞を付与した．テストデータに対しては，全ての学習データで訓練した Stanford POS Tagger を用いて各単語の品詞を決定した．

提案モデルのハイパーパラメータは，表 1 のように設定した．また，モデル学習に用いる確率的勾配法の学習率 λ は，$\lambda_t = \frac{0.0075}{t}$ とした．ただし，t はイテレーション数である．単語の埋め込みベクトル e^{word} は，word2vec[*2] を用いて，英語の Gigaword コーパス [*3] に含まれる NewYork Times データから，100 次元のベクトルを事前に学習した．

4.2 実験結果

表 2，表 3 に実験結果を示す．表 2 は，テストデータの全トークンに対する複単語表現解析の性能を示している．評価は完全一致基準に基づいており，正解のトークンと予測したトークンの範囲が完全に一致し，かつ品詞も正しいときにのみ正解とする．

表 2 に示されているように，提案手法が最も高い性能を達成していることがわかる．ただし，データに含まれる複単語表現の割合はあまり多くないため，評価の大部分は単一の単語に対する品詞タグ付けの精度に依存している．したがって，提案手法は，現在広く用いられている Stanford POS Tagger と比べても同等以上の精度で単語の品詞タグ付けを行えていることを示している．

次に，表 3 は，単一の単語を評価から取り除き，2 単語以上の複単語表現のみの解析性能を評価した結果である．提案手法は，複単語表現のみの評価でも他手法を上回ることがわかった．これは，提案モデルが文字や単語，またはその文脈から複単語表現の特徴量を学習できており，高精度で複単語表現の同定が行えることを示している．拡張 BIO 系列ラベリングの手法 [10] は，1 単語ずつ BIO タグ付けを行っていくのに対して，提案手法は，複単語表現を構成する全ての単語を一度に考慮して判別を行うため性能向上につながったのではないかと考えられる．

[*1] http://nlp.stanford.edu/software/tagger.shtml
[*2] https://code.google.com/p/word2vec/
[*3] https://catalog.ldc.upenn.edu/LDC2011T07

5. おわりに

本稿では，畳み込みニューラルネットワークを用いて文字，単語，複単語表現の特徴量をデータから自動的に学習し，文に含まれる複単語表現の同定および品詞タグ付けを行う手法を提案した．提案手法は，句動詞のように単語が文中で連続していない場合にも頑健に複単語表現を同定することができ，規則ベースや系列ラベリングに基づく手法よりも高性能であることを実験的に確認した．今後は，複単語表現の解析と，構文解析を統合するモデルや手法を考案する予定である．

参考文献

[1] Carpuat, M. and Diab, M.: Task-based Evaluation of Multiword Expressions : a Pilot Study in Statistical Machine Translation, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pp. 242–245 (2010).

[2] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. and Kuksa, P.: Natural Language Processing (almost) from Scratch, *Journal of Machine Learning Research*, Vol. 12, No. Aug, pp. 2493–2537 (2011).

[3] Constant, M. and Sigogne, A.: MWU-Aware Part-of-Speech Tagging with a CRF Model and Lexical Resources, *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pp. 49–56 (2011).

[4] dos Santos, C. N. and Bianca, Z.: Learning Character-level Representations for Part-of-Speech Tagging, *Proceedings of the 31st International Conference on Machine Learning*, pp. 1818–1826 (2014).

[5] Green, S., de Marneffe, M.-C. and Manning, C. D.: Parsing Models for Identifying Multiword Expressions, *Computational Linguistics*, Vol. 39, pp. 195–227 (2013).

[6] Hogan, D., Cafferkey, C., Cahill, A. and van Genabith, J.: Exploiting Multi-Word Units in History-Based Probabilistic Generation, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 267–276 (2007).

[7] Nivre, J. and Nilsson, J.: Multiword Units in Syntactic Parsing, *Workshop on Methodologies and Evaluation of Multiword Units in Real-World Applications (MEMURA)*, pp. 1–8 (2004).

[8] Pichotta, K. and DeNero, J.: Identifying Phrasal Verbs Using Many Bilingual Corpora, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 636–646 (2013).

[9] Sag, I. A., Baldwin, T., Bond, F., Copestake, A. and Flickinger, D.: Multiword Expressions: A Pain in the Neck for NLP, *Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CONLL)*, pp. 1–15 (2001).

[10] Schneider, N., Danchik, E., Dyer, C. and Smith, N.: Discriminative Lexical Semantic Segmentation with Gaps: Running the MWE Gamut, *Transactions of the Association for Computational Linguistics*, Vol. 2, pp. 193–206 (2014).

[11] Schneider, N., Onuffer, S., Kazour, N., Danchik, E., Mordowanec, M. T., Conrad, H. and Smith, N. A.: Comprehensive Annotation of Multiword Expressions in a Social Web Corpus, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pp. 455–461 (2014).
[12] Shigeto, Y., Azuma, A., Hisamoto, S., Kondo, S., Kouse, T., Sakaguchi, K., Yoshimoto, A., Yung, F. and Matsumoto, Y.: Construction of English MWE Dictionary and its Application to POS Tagging, *Proceedings of the 9th Workshop on Multiword Expressions*, pp. 139–144 (2013).
[13] Tjong Kim Sang, E. F. and De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, *Proceedings of Conference on Computational Natural Language Learning (CoNLL)*, pp. 142–147 (2003).
[14] Tratz, S. and Hovy, E.: A Taxonomy, Dataset, and Classifier for Automatic Noun Compound Interpretation, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 678–687 (2010).
[15] 駒井雅之，進藤裕之，松本裕治：英語の句動詞表現の同定とコーパス構築，言語処理学会第 21 回年次大会，pp. 744–747 (2015).