

## ニューラルネットワークを利用した中国語の統合的な構文解析

栗田 修平<sup>†</sup>・河原 大輔<sup>†</sup>・黒橋 禎夫<sup>†</sup>

ニューラルネットワークに基づく係り受け解析モデルは、近年の深層学習を利用した言語処理研究の中でも大きな潮流となっている。しかしながら、こうした係り受け解析モデルを中国語などの言語に適用した際には、パイプラインモデルとして同時に用いられる単語分割や品詞タグ付けモデルの無視できない誤りによって性能が伸び悩む問題が存在する。これに対しては、単語分割・品詞タグ付けと係り受け解析の統合モデルを利用し、単語分割と構文木作成とを同時に行うことでその双方の改善が期待される。加えて、中国語においては個々の文字が固有の意味を持ち、構文解析では、文字やその組み合わせである文字列もしくは部分単語の情報が単語単位の情報と並んで本質的な役割を果たすことが期待される。本研究では、ニューラルネットワークに基づいて、単語分割と品詞タグ付け、もしくは単語分割と品詞タグ付け、係り受け解析の統合構文解析を行うモデルを提案する。また、同時に、文字列や部分単語の情報を捉えるために、文字や単語の分散表現に加えて、文字列の分散表現を利用する。

キーワード：構文解析、係り受け解析、遷移に基づく解析、深層学習、単語の分散表現

## Neural Network-based Chinese Joint Syntactic Analysis

SHUHEI KURITA<sup>†</sup>, DAISUKE KAWAHARA<sup>†</sup> and SADAO KUHASHI<sup>†</sup>

Recently, dependency parsers with neural networks have outperformed existing parsers. When these parsing models are applied to Chinese sentences, they are used in a pipeline model with word segmentation and POS tagging models. In such cases, parsing models do not work well because of word segmentation and POS tagging errors. This can be solved by joint models of word segmentation, POS tagging and dependency parsing. In addition to this, Chinese characters have their own meanings, so the meanings of characters, character strings and sub-words are as important as the meanings of words in dependency parsing. In this study, we propose a neural network-based joint word-segmentation, POS tagging and dependency parsing model in addition to a joint word-segmentation and POS tagging model. We exploit not only word and character embeddings but also character string embeddings in all our models.

**Key Words:** *Syntactic Analysis, Dependency Parsing, Transition-based Parsing, Deep Learning, Word Embeddings*

---

<sup>†</sup> 京都大学大学院情報学研究所, Graduate School of Informatics, Kyoto University, JST CREST

## 1 はじめに

近年、ニューラルネットワーク及び分散表現の使用により、係り受け解析は大きく発展している。(Chen and Manning 2014; Weiss, Alberti, Collins, and Petrov 2015; Zhou, Zhang, Huang, and Chen 2015; Alberti, Weiss, Coppola, and Petrov 2015; Andor, Alberti, Weiss, Severyn, Presta, Ganchev, Petrov, and Collins 2016; Dyer, Ballesteros, Ling, Matthews, and Smith 2015). こうした構文解析器が、単語ごとの分かち書きを行う英語や多くのヨーロッパ諸語に適用された場合は非常に正確に動作する。しかし、日本語や中国語のように、特に単語毎の分かち書きを行わない言語に対し適用する場合は、事前に形態素解析器や単語分割器を利用して単語分割を行う必要がある。また、単語分割が比較的容易な言語の場合でも、構文解析器は品詞タグ付け結果を利用することが多い。したがって、前段の単語分割器や品詞タグ付け器と後段の構文解析器をパイプラインにより結合されて用いられる。しかし、どのような単語分割器や品詞タグ付け器にも出力の誤りが存在し、結果的にそれが後方の係り受け解析器にも伝播することで、全体の解析結果が悪くなってしまう問題が存在した。これを誤差伝播問題と呼ぶ。日本語においても中国語においても、単語の定義には曖昧性が存在するが、特に中国語では、このような単語の定義の曖昧性から、単語分割が悪名高く難しいことが知られている (Shen, Kawahara, and Kurohashi 2016a). それゆえ、従来法である単語分割、品詞タグ付け、構文解析のパイプラインモデルは、単語分割の誤りに常に悩まされることになった。単語分割器が単語の境界を誤って分割してしまうと、伝統的な one-hot な単語素性や通常の単語の分散表現 (**word embedding**) では、もとの単語の意味を正しく捉えなおすことは難しい。結果的に、中国語の文を生文から解析する際は、パイプラインモデルの精度は 70%前半程度となっていた (Hatori, Matsuzaki, Miyao, and Tsujii 2012).

このような誤差伝播問題に対しては、統合モデルを使用することが有効な解決方法として提案されている (Zhang and Clark 2008, 2010; Hatori, Matsuzaki, Miyao, and Tsujii 2011; Hatori et al. 2012; Zhang, Zhang, Che, and Liu 2014). 中国語の単語は、単一の表層系で複数の構文的な役割を演じる。ゆえに、そうした単語の境界を定めることと、後続の品詞タグ付け、構文解析は非常に関連のあるタスクとなり、それらを別個に行うよりも、同時に処理することで性能の向上が見込まれる。中国語の統合構文解析器については、すでに Hatori et al. (2012) や Zhang et al. (2014) などの統合モデルが存在する。しかし、これらのモデルは、近年の word embedding のような表現学習や、深層学習手法を利用しておらず、専ら、複雑な素性選択や、それら素性同士の組み合わせに依存している。本研究では、ニューラルネットワークを用いた手法による中国語の統合構文解析モデルを提案し、パイプラインを用いたモデルとも比較する。

ニューラルネットワークに基づく係り受け解析では、単語の分散表現と同様に文字の分散表現が有効であることが英語などの言語における実験で示されている (Ballesteros, Dyer, and Smith 2015). しかし、中国語や日本語のように個々の文字が固有の意味を持つ言語において、単語以下の構造で

ある部分単語の分散表現がどのように有効であるかについては、いまだ十分な研究が行われていない。中国語では単語そのものの定義がやや曖昧である他に、単語内にも意味を持つ部分単語が存在する場合がある。加えて、中国語の統合構文解析を行う場合には、単語分割の誤りに対処したり、文中で単語分割をまだ行っていない箇所の先読みを行う必要があり、必然的に、単語だけではなく部分単語や単語とはならない文字列の意味を捉えることが必要になる。このような部分単語や単語とはならない文字列は、大抵の場合はモデルの学習に用いる訓練コーパスや事前学習された単語の分散表現中には存在せず、文字や文字列の分散表現を扱わない先行研究では未知語として処理される。しかし、こうした文字列を未知語として置換し処理するよりも、その構成文字から可能な限りその意味を汲み取った方が、より高精度な構文解析が行えると考えられる。このため、本研究では文字列の分散表現を利用した統合構文解析モデルを提案する。提案手法では、既知の文字または単語についてはそれらの分散表現を使用し、未知の文字列については文字列の分散表現を使用する。

本研究では中国語の統合構文解析モデルとして、単語分割・品詞タグ付けおよび係り受け解析の統合モデルと、単語分割と品詞タグ付けの統合モデルおよび係り受け解析のパイプラインモデルの2つを提案する。これらのモデルを使用することで、実験では新規に世界最高性能の中国語単語分割および品詞タグ付け精度を達成した。また、係り受け解析とのパイプラインモデルが、従前の統合解析モデルと比較して、より優れた性能を達成した。以上の全てのモデルにおいて、単語と文字の分散表現に加えて文字列の分散表現を利用した。著者の知る限りにおいて、これは分散表現とニューラルネットワークを利用し、中国語の単語分割・品詞タグ付け・係り受け解析の統合解析を行った、はじめてのモデルである。

この論文における貢献は以下のようにまとめられる。(1) 分散表現に基づく、初めての統合構文解析モデルを提案した。(2) 文字列の分散表現を未知語や不完全な文字列に対してその意味を可能な限り汲み取るために使用した。(3) 加えて、既存手法で見られた複雑な素性選択を避けるために、双方向 LSTM を使用するモデルを提案した。(4) 中国語のコーパスにおける実験で単語分割・品詞タグ付けおよび係り受け解析にて新規に世界最高性能を達成した。

この他に、本論文では中国語係り受け解析のラベル付けモデルを提案し、原文からラベル付き係り受け解析までを行った際のスコアを評価する。このモデルに関しても、同様に文字列の分散表現を利用する。

## 2 関連研究

遷移型係り受け解析アルゴリズムは、係り受け解析を行う際にその精度の高さと処理速度の速さから広く使われているモデルである。特に中国語の統合遷移型係り受け解析に対するアルゴリズムは Hatori et al. (2012) および Zhang et al. (2014) において提案された。Hatori et al. (2012) は、係り受け解析に関する情報が、単語分割と品詞タグ付けを行う際にも有効であることを示し、遷移

型係り受け解析アルゴリズムに基づく最初の単語分割、品詞タグ付けおよび係り受け解析のモデルを提案した。Zhang et al. (2014) はこれを拡張し、通常の単語間の係り受けに加えて、単語内部の係り受けに対して独自にアノテーションを施し、これを考慮することによって、精度の向上を達成できることを示した。

こうした Hatori et al. (2012) および Zhang et al. (2014) による提案手法により、パイプラインモデルより優れた性能を出すなど、大きな成果が得られた。一方で、このような従来法におけるモデルでは、文字および単語の one-hot な表現を専ら利用しており、使用する文字や単語の類似性は考慮されていなかった。加えて、中国語の統合構文解析により処理を行う最中には、素性として、既知の単語や文字だけではなく単語をなさない不完全な文字列や部分単語も出現する。そのような不完全な文字列もしくは未知の単語は、モデルになんらかの形で認識させることで構文解析を行う際の重要な情報となりうる。しかし、そのような不完全な文字列は、従来法における外部知識源である辞書情報や、学習済みの通常の単語の分散表現では捉えることが難しかった。

Hatori et al. (2012) および Zhang et al. (2014) の提案手法における他の問題点として、これらの先行研究が詳細な素性選択を用いていることがあげられる。一方で、近年、双方向 LSTM を利用し詳細な素性選択を回避したニューラルネットワークモデルが提案されている (Kiperwasser and Goldberg 2016; Cross and Huang 2016)。これらのモデルでは、双方向 LSTM は語句の分散表現をその内容も含めてモデル化するために用いられている。実際に、Kiperwasser and Goldberg (2016) による構文解析モデルにおいては、双方向 LSTM により文全体の情報を利用することができるのに対し、素性に基づくモデルでは文全体を素性として用いることはできない。また、このように双方向 LSTM を利用して文全体を符号化する手法は、機械翻訳でも多く用いられている (Bahdanau, Cho, and Bengio 2014)。結果として、Kiperwasser and Goldberg (2016) によるモデルでは世界最高の性能と比較可能な性能を達成している。本研究では、 $n$  グラムの文字分散表現に対応した双方向 LSTM を利用した構文解析モデルも提案する。

なお、Hatori et al. (2012) による手法ではあくまでも単一の分類器を使用して単語分割・品詞タグ付けおよび係り受け解析を同時に行うのに対し、Zhang et al. (2014) の提案手法では、単語分割・品詞タグ付けに対応する分類器と係り受け解析に対応する分類器を同時に用意し、単語分割・品詞タグ付けに対応する分類器を数単語分だけ先に処理させる。このような2つの手法は、本研究における単語分割・品詞タグ付けおよび係り受け解析の統合モデルと、単語分割・品詞タグ付けの統合モデルおよび係り受け解析のパイプラインモデルの関係に近い。4.2.3 節にて、この関係を詳細に議論する。

### 3 モデル

この論文で提案する統合構文解析モデルは, 3.1 節で解説する遷移型アルゴリズムを使用する. また, 不完全な文字列に対する分散表現を得るための文字列の分散表現手法については 3.2 節で解説する. 素性選択を利用するニューラルネットワークモデルを 3.3 節で解説し, 双方向 LSTM を利用するニューラルネットワークモデルを 3.4 節で解説する.

#### 3.1 単語分割・品詞タグ付け, 係り受け解析のための遷移型アルゴリズム

単語分割・品詞タグ付け, 係り受け解析のための統合アルゴリズムには Hatori et al. (2012) に基づく, 遷移型の統合アルゴリズムを使用する. 統合解析モデルは入力文字列からなる buffer と stack からなり, buffer は文字が格納され, stack には単語と品詞タグ, およびそれらの単語の構文木上の子孫に当たる単語が格納される (図 1). 初期状態では buffer に全ての入力文字列が格納される. stack は空である. また, アルゴリズムの便宜上, buffer の末尾には文末を意味するシンボル “EOS” を置く. 下記の遷移操作に従って, buffer から stack に文字が移され, 単語や構文木が形成される. 終状態においては, buffer が空になり, stack には EOS シンボルを係り受け解析における ROOT ノードとして, 品詞タグが付与された単語群による構文木が生成されている.

本研究で用いる遷移操作は以下の通りである.

- SH (*t*) (*shift*): buffer の最初の文字を stack の先頭へ移動させ, 新しい単語とする
- AP (*append*): buffer の最初の文字を stack の先頭の単語の末尾に加える
- RR (*reduce-right*): stack の先頭 2 単語のうち, 右側の単語を stack から消去し, 左側の単語の右側の子供の単語とする
- RL (*reduce-left*): stack の先頭 2 単語のうち, 左側の単語を stack から消去し, 右側の単語の左側の子供の単語とする

SH 操作および AP 操作によって単語の境界と品詞タグが決定され, RR 操作および RL 操作により

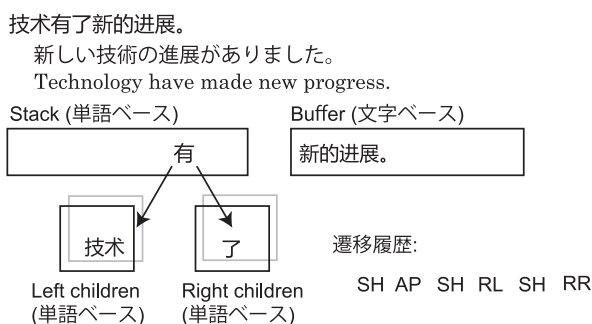


図 1 単語分割・品詞タグ付け, 素性選択のための遷移に基づく統合アルゴリズム

構文木が作成される。これらの遷移操作のうち、RR 操作および RL 操作は Arc-standard アルゴリズムのものと同一である (Nivre 2004)。SH 操作および AP 操作はいずれも buffer の最初の文字を stack へと移動させるが、SH 操作は移動した文字を新しい単語の先頭とするのに対し、AP 操作はすでに stack 先頭にある部分単語の末尾に文字を加える。ゆえに、SH 操作はモデルが予測する単語数を 1 単語だけ増加させるのに対し、AP 操作は、stack の先頭に存在する部分単語を 1 文字分だけ延長させる。品詞タグは SH( $t$ ) 操作と同時に付与される。ゆえに、SH( $t$ ) 操作は品詞タグの数と同じだけ存在する。

なお、buffer 上に文字が存在しない時は、SH 操作および AP 操作を行うことはできない。同様に stack 上に 2 個以上の単語または生成途中の単語が存在しない時は RR 操作および RL 操作を行うことはできない。また、AP 操作が適用可能となるのは、直前の操作が SH 操作、もしくは AP 操作である時のみである。

この論文においては、貪欲法による訓練及び探索とビーム法による訓練および探索の両方を実験する。この解析アルゴリズムはその両方に対して動作する。また、本研究においては、単語分割と品詞タグ付けの統合モデルと係り受け解析の単独モデルも同様に作成した。単語分割と品詞タグ付けの統合モデルにおいては、RR 操作および RL 操作は用いない。同様に、単独の係り受け解析モデルにおいては、通常の Arc-standard 構文解析モデルを使用する。

### 3.2 単語および文字、文字列の分散表現

本節ではニューラルネットワークモデルにおいて用いられる分散表現について解説する。

統合構文解析を行う際は、必然的に、意味のある単語と不完全な文字列が解析中に多く出現する。このような不完全な文字列の表現は、Hatori et al. (2012) などの既存手法においては用いられなかったが、もし、分散表現を用いて他の単語や文字列との類似性を表現できれば、統合構文解析を行う際に非常によい情報となりうるはずである。

例えば、固有表現である“南京东路”（上海にある有名な商店街）は Penn Chinese Treebank (CTB) コーパスではひとつの単語として扱われている。このような固有表現は、“北京西路”や“湘西路”などとして CTB コーパス中に多く出現する。これらは、“南京”や“北京”などの都市を表す部分単語と“东路”や“西路”などといった場所を表す部分単語の組み合わせとして理解される。こうした表現を学習することで、テストデータ中に未知の“(固有地名) 东路”や“(固有地名) 西路”といった表現が出現しても、これを地名として処理することが期待できる。

また、部分単語や、逆に複数の単語にまたがる文字列を利用することが、単語分割の誤りを補うために役立つ場合もある。例えば、“南京东路”に対し、単語分割器が過剰に単語分割を行い、“南京”と“东路”の 2 単語に区切ってしまった場合を仮定しよう<sup>1</sup>。この時、事前学習された単語の分

<sup>1</sup> アノテーション基準によっては、そもそもこのような分割の仕方が正しい場合もある。

散表現や CTB の訓練コーパス中に“东路”という単語が存在しなければ、これは未知語として処理されうる。このような場合でも、その文字の構成から“东路”は場所を表す単語であると推測できる。このように、部分単語や文字の表現を使用することで、単語分割の誤りに対して頑健さを持たせることができる。

しかしながら、こうした部分単語や文字列等は、多くの従来研究では未知語もしくは“UNK”シンボル等として置換され、処理されていた。本研究では、こうした文字列についても、それを構成する文字から可能な限り意味を捉える分散表現を提案する。 $n$  個の文字  $c_i$  からなる文字列  $c_1c_2\cdots c_n$  を考える。提案手法ではこの文字列に対する分散表現  $\mathbf{v}(\cdot)$  を以下のように計算する。

$$\mathbf{v}(c_1c_2\cdots c_n) = \sum_{i=1}^n \mathbf{v}_c(c_i) \quad (1)$$

ここで  $\mathbf{v}_c(\cdot)$  は文字の分散表現ベクトルを表す。

このように文字列の分散表現を、その構成文字の文字表現から構成する手法は、どのような文字列に対しても計算できるほか RNN や CNN 等の複雑なニューラルネットワークを用いないことによる速度上の利点が存在する。

なお、このような文字、単語および文字列の分散表現はすべて同じベクトルの次元を持ち、ニューラルネットワークの計算グラフ中で、入力に対しどのような分散表現を利用するかが決定される。したがって、いずれの分散表現を利用する場合でも、その基となる文字または単語の分散表現が、ニューラルネットワークの誤差逆伝播法により学習される。

本研究では、任意の中国語の単語列に対し、以下の規則にて、その分散表現を割り当てる。

- (1) 文字列が既存の単語分散表現中に存在するか調べ、存在する場合には、その分散表現を使用する。
- (2) 分散表現が存在しない場合には、上記の式のように、文字に対する分散表現を用いて、文字列の分散表現として使用する。
- (3) 分散表現中に存在しない文字については UNK シンボルを使用する。

本研究では、UNK シンボルのような未知語の表現ベクトルを利用することは可能な限り回避した。なぜならば、そのようなベクトルを使用することはニューラルネットワークに与える入力を縮退させることになるからである。しかしながら、事前学習された分散表現中に存在しない文字については、入力を UNK シンボルに置換して使用する。

品詞タグについても同様に対応する分散表現を用意し、学習に使用する。UNK シンボルに対応する分散表現や品詞タグに対応する分散表現は、正規分布を用いて初期化される。その他、本研究では、部分単語の長さに対応する、数字の分散表現を一部に用いた。これは、一定の長さを持つ文字列は、同じ分散表現を共有するものである。もっぱら、buffer から stack へと一文字毎に移されている最中の部分単語の長さを表現するために用いられる。数字の分散表現も正規分布を用いて初

期化される。

提案手法では、単語と文字の分散表現を事前学習にて準備する。文字と単語の分散表現は、同一のベクトル空間に埋め込まれる。まず、事前学習に用いるコーパスについて、単語分割されたファイルと一文ごとに文字分割されたファイルを用意する。各ファイルには一行に一文ずつ事前学習に用いる文を配置する。次に、それらのファイルを結合し、行単位でランダムに並び替え、単語及び文字の分散表現を学習するための事前学習コーパスとする。これにより、類似した意味を持つ単語と中国語の文字は、ベクトル空間内の近い位置に埋め込まれることが期待される。中国語の文中には、一文字で一単語となる単語も存在し、そのような単語を介して、周辺の文字や単語が類似した位置に配置されうるからである。

具体的な分散表現学習に用いたツールおよびコーパスについては、4.1 節に記載した。

### 3.3 素性選択を利用するニューラルネットワークモデル

#### 3.3.1 ニューラルネットワークモデル

素性選択に基づくニューラルネットワークモデルを図 2 に示す。素性とその分散表現を入力に取るニューラルネットワークを使用した構文解析は、近年、活発な研究が行われており、その中には貪欲法に基づいてニューラルネットワークを学習させるもの (Chen and Manning 2014; Weiss et al. 2015)、ビーム探索を用いるもの (Andor et al. 2016; Weiss et al. 2015) がある。本研究では、このように素性を入力として用いることに加えて、不完全な文字列が素性として入力されても処理が可能であるように、3.2 節にて導入した文字列の分散表現の動的生成を加えた。

遷移操作を決定する分類器には、8,000 次元の 2 つの隠れ層を持つ、3 層フィードフォワードニューラルネットワークを使用した。これは 200 次元の隠れ層を利用した (Chen and Manning 2014) や 1,024 次元もしくは 2,048 次元の隠れ層を利用した Weiss et al. (2015) よりも更に大きい。隠れ層の活性化関数には ReLU を使用した (Nair and Hinton 2010)。なお、(Chen and Manning 2014) にて用いられている  $x^3$  の活性化関数は、2 層のフィードフォワードニューラルネットワークでは比較可能な性能を達成したが、3 層以上のフィードフォワードニューラルネットワークでは、うまく動作しなかった。フィードフォワードニューラルネットワークの最終層が出力層となる。貪欲法による出力層は softmax 関数を使用した。ニューラルネットワークの隠れ層は乱数を用いて初期化した。貪欲法における損失関数  $L(\theta)$  は

$$L(\theta) = - \sum_{s,t} \log p_{s,t}^{\text{greedy}} + \frac{\lambda}{2} \|\theta\|^2,$$

$$p_{s,t}^{\text{greedy}}(\beta) \propto \exp \left( \sum_j w_{tj} \beta_j + b_t \right),$$

となる。ここで、 $t$  は可能な遷移操作の集合  $\mathcal{T}$  ( $t \in \mathcal{T}$ ) の中のある 1 つの遷移操作を表す。  $s$  は



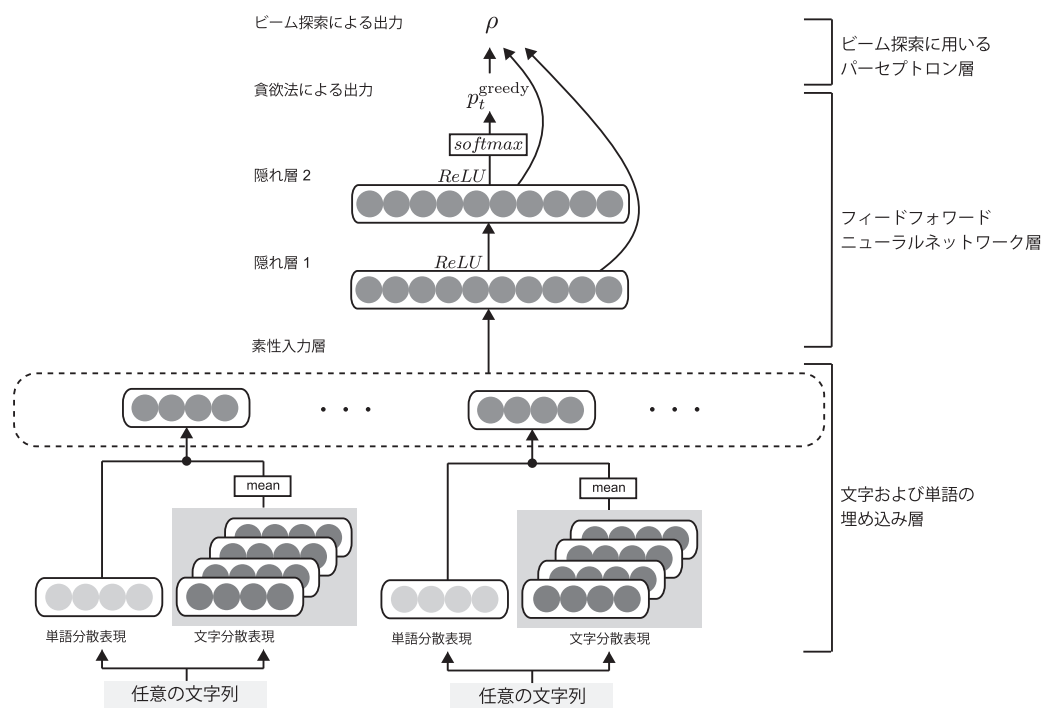


図 2 素性に基づくニューラルネットワーク. 貪欲法による出力は 2 番目の隠れ層から入力を受けるのに対し, ビーム探索には全ての隠れ層および貪欲法による出力層から入力を受けるパーセプトロン層を使用する. 入力の文字列は, 分散表現に含まれている場合はその分散表現が使用され, 含まれていない場合は, 文字列の分散表現がその文字構成から計算されて使用される.

ニューラルネットワークの学習におけるミニバッチ中の 1 つの要素を示す.

$\beta$  はフィードフォワードニューラルネットワーク最終層における出力を表す.  $w_{tj}$  及び  $b_t$  はニューラルネットワークの重み行列とバイアス項を示す.  $\theta$  はモデルの全ての変数を示す. 本研究ではニューラルネットワークの変数に対し, L2 罰則項と Dropout を使用した. ニューラルネットワークの誤差逆伝播法は, これらのニューラルネットワークの変数を含めて, 単語及び文字の分散表現まで行われる. 誤差逆伝播法の学習率の調整には Adagrad を使用する (Duchi, Hazan, and Singer 2010). Adam (Kingma and Ba 2015) と SGD を使用することも考慮したが, このモデルについては Adagrad がよりよく振る舞った. 他のモデル変数については表 1 にまとめる.

表 1 のモデル変数のうち,  $\mathbf{h}_1$  と  $\mathbf{h}_2$  のサイズは予備実験により決められた. 学習率の初期値については, ニューラルネットワークの学習において広く使われているものを用いたが, ビームを使用する場合は, モデルの安定性のため, 小さな初期値から学習を始めた. 単語・文字の埋め込みにおけるボキャブラリー数については, 分散表現に含まれる語彙数を多くするため, サイズの大きいものを用いた. 埋め込みの次元は Chen and Manning (2014) などの先行研究で広く使われている値

表 1 ニューラルネットワークの構造と学習に関する諸変数

Type	Value
$h_1$ と $h_2$ のサイズ	8,000
学習率の初期値	0.01
ビーム探索における学習の初期値	0.001
単語・文字埋め込み層の語彙数	1 M
単語・文字埋め込みのサイズ	200
文字列長さの埋め込みのサイズ	20
ミニバッチサイズ	200

単語と文字は同じ次元の分散表現を持つ。「文字列長さの埋め込み」とは、buffer から stack へと一文字毎に移されている最中の部分単語の長さに対応する数字の分散表現である。

の一つである 200 次元を用いた。

ニューラルネットワークの隠れ層が 8,000 次元となり、これは先行研究よりも大きい。この理由として、1 つ目に、3.3.2 節にて紹介する、素性を表現するベクトル表現が大きいことが挙げられる。例えば、単語分割、品詞タグ付けおよび係り受け解析の統合モデルについては、合計で 9,820 次元にもなる。この中には文字列に関する素性も含まれる。これを表現するために大きな隠れ層が必要となる。2 つ目に、ニューラルネットワークが、単語分割、品詞タグ付けおよび係り受け解析という多岐にわたる判断をしなければならないことが挙げられる。出力側が複雑になるために、必要となる隠れ層のサイズも大きくなると考えられる。

貪欲法によるモデルの学習では、学習前に、訓練データセット中の全ての文に対して、遷移型構文解析アルゴリズムを正解ラベルに従って順次適用させたときに出現する、入力素性および遷移操作の教師ラベルの組み合わせを予め計算し抽出した。このようにして抽出された文内での遷移操作と入力素性の組み合わせを、文をまたいでシャッフルしてからニューラルネットワークの学習に用いるミニバッチを作成した。これにより、あるミニバッチに特定の文が偏ることがないようにした。

モデルのテスト時や実際の入力文の解析時には、処理を行う文を複数個並べてミニバッチを作成し、同時に解析が行えるようにした。これにより、ある一連のニューラルネットワークの呼び出しに対し、複数文を同時に処理することが可能となり、速度の大幅な向上が可能となった。この際に、同時に処理が可能な文数は専ら GPU のメモリサイズのみにより制約される。また、この方法はビームを用いた解析にも適用できる。

### 3.3.2 素性

このニューラルネットワークの素性を表 2 に列挙する。本研究では、以下の 3 種類の素性を使用した。

- (1) Hatori et al. (2012) にて用いられた素性から、素性同士の共起を取り除いたもの

表 2 統合モデルに対する素性

Type	Features
Stack word and tags	s0w, s1w, s2w s0p, s1p, s2p
Stack 1 children and tags	s0l0w, s0r0w, s0l1w, s0r1w s0l0p, s0r0p, s0l1p, s0r1p
Stack 2 children	s1l0w, s1r0w, s1l1w, s1r1w
Children of children	s0l0lw, s0r0rw, s1l0lw, s1r0rw
Buffer characters	b0c, b1c, b2c, b3c
Previously shifted words	q0w, q1w
Previously shifted tags	q0p, q1p
Character of q0	q0e
Parts of q0 word	q0f1, q0f2, q0f3
Strings across q0 and buf.	q0f1b1, q0f1b2, q0f1b3
Strings of buffer characters	b0-2, b0-3, b0-4 b1-3, b1-4, b1-5 b2-4, b2-5, b2-6 b3-5, b3-6 b4-6
Length of q0	lenq0

この表において, “b” は buffer の文字, “s” は stack の単語を示し, “b” と “s” の次の数字は, 先頭からの文字および単語の番号である. “l” と “r” は, その単語の左側と右側の子どもの単語のことを指し, “l” と “r” の次の数字は, 子どもの単語の番号である. サフィックスの w, p, c, e は, それぞれ, 文字列または単語, 品詞タグ, 文字, そして文字列の最後の一文字を表す. “q0” は直前に buffer から stack へ移動された単語を, “q1” は “q0” より先に移動された単語を示す. “part of q0 word” では, q0f1, q0f2, q0f3 は, それぞれ f1, f2, f3 の位置の文字から始まる q0 の部分単語を表す. ここで f1 は q0 の最後の文字の位置を表すため, q0f1 は q0 の最後の一文字を表す. f2 は最後から 2 番目の文字の位置を表すため, q0f2 は f2 の位置から始まる 2 文字の文字列を示す. “strings across q0 and buf.” では, q0f1bX は, 単語 q0 の f1 の位置から始まる X 個の文字からなる文字列を表す. この素性による文字列は buffer と stack にまたがりうるが, 連続的な文字列に限定する. “strings of buffer characters” では, bX-Y が, buffer 内部の X 番目から Y 番目までの文字列を表す. lenq0 は q0 の長さを示し, この分散表現のみ小さな埋め込みのサイズを用いる.

(2) Chen and Manning (2014) にて用いられた素性

(3) 文字列に関する独自の素性

このなかで独自の素性として, 部分単語に関するものと buffer と stack にまたがる文字列の素性, buffer 中の文字列の素性が存在する. buffer と stack にまたがる文字列の素性は, 現在, 単語分割が行われている単語に対する貴重な情報を提供する. また, 長すぎる文字列に対する素性を用いることがないように, 本研究では文字列素性の長さを 4 文字以下に制限した. CTB データセットにおいては, 5 文字以上の長さの単語は稀であるからである. ただし, これは素性に関するものであり, 出力単語としてはそれ以上の長さのものもありうる. ただし, 本研究では, 文中に連続に出

現する文字列のみを素性に用い、共起に関する素性は使用しなかった。ここでいう共起に関する素性とは、例えば stack の先頭以外の位置にある文字列と、buffer にある文字列の組み合わせに関する素性等である。Hatori et al. (2012) では多数の手作業で調整された one-hot な素性の共起を利用していた。本研究では、そのような素性同士の共起はニューラルネットワークの隠れ層上で自然に考慮されると考えた (Hinton, McClelland, and Rumelhart 1986)。

本研究では、単語分割済みの単語については、単語や文字列のほかに、すでにモデルによって付与された品詞も素性として利用させた。また、buffer から stack へと一文字ずつ文字を移動させている最中の単語の長さを表現するために、小さな次元の埋め込みを用いた。

また、単語分割、品詞タグ付けおよび係り受け解析の統合解析モデルとは別に、単語分割と品詞タグ付けのみの統合モデルも実験に用いた。このモデルについては、構文解析に関する素性、すなわち stack に存在する単語や文字列の子供や孫にあたる単語に関する素性を省略している。

### 3.3.3 ビーム探索

ビーム探索は中国語の統合解析において重要な役割を果たす。本研究では、近年提案された、ビームを用いたニューラルネットワークの最適化手法を採用した (Weiss et al. 2015; Andor et al. 2016)。図 2 では、最上部に位置するパーセプトロン層がビームを利用した解析に使用する。具体的には、ビーム探索のために独立したパーセプトロン層を用意し、フィードフォワードニューラルネットワークの 2 層の隠れ層および貪欲法による出力層を、その入力とする:  $[\mathbf{h}_1, \mathbf{h}_2, \mathbf{p}^{\text{greedy}}(\mathbf{y})]$ 。次に、このパーセプトロン層を以下のコスト関数にしたがって学習させる (Andor et al. 2016):

$$L(d_{1:j}^*; \theta) = - \sum_{i=1}^j \rho(d_{1:i-1}^*, d_i^*; \theta) + \ln \sum_{d_{1:j} \in \mathcal{B}_{1:j}} \exp \sum_{i=1}^j \rho(d'_{1:i-1}, d'_i; \theta),$$

ここにおいて、 $d_{1:j}$  は統合構文解析の遷移経路を示し、 $d_{1:j}^*$  は正解の遷移経路を示す。 $\mathcal{B}_{1:j}$  はビーム内部での 1 番目から  $j$  番目までの遷移経路を示す。 $\rho$  は図 2 におけるパーセプトロン層の出力を表す。

本研究では、Andor et al. (2016) と同様にして、貪欲法を用いてフィードフォワードニューラルネットワーク部分を学習し、次にビームを用いて、パーセプトロン層のみを学習させる。最後に、フィードフォワードニューラルネットワークを含めて誤差逆伝播法による学習を行う。この学習はネットワーク全体に対して行うことも可能であるが、予備実験の結果、この段階で文字および単語の埋め込み層を最適化すると、学習結果が悪くなることが判明した。そこで、本研究では、文字及び単語埋め込み層をこの段階の学習からは除外した。このニューラルネットワーク全体を通した誤差逆伝播法は、かなりの GPU メモリを消費する。それゆえ、訓練コーパスのうちとりわけ長い文

は、GPU メモリに載せられないために訓練から取り除いた。また、ビームを用いた訓練においては、最初のエポックでは小さなビームサイズで学習を行い、徐々に学習に用いるビームサイズを増やしていった。学習には順に 4, 8, 16 のサイズのビームを使用した。最終的な単語分割・品詞タグ付けおよび係り受け解析の統合モデルの学習及びテストにはサイズ 16 のビームを用いた。

本研究では、単語分割・品詞タグ付けおよび係り受け解析の統合モデルでのビーム探索について、Hatori et al. (2012) と同様に特別な配置ステップを使用した。単語分割・品詞タグ付けおよび係り受け解析の統合モデルでのビーム探索では、AP 遷移のみに 2 ステップの配置をもたせている。この配置を使用することで、 $N$  文字の文に対する総遷移回数は、どのような遷移経路を経由しても、文終端記号に対応する遷移を除いて  $2N - 1$  となる。これは、AP 遷移操作が、文字の追加と単語内の係り受け解決の 2 つのを行っているとして解釈できる。つまり、中国語の単語は、単語内にも文字同士の係り受け関係が存在すると解釈し、AP 遷移操作はこの単語内の係り受けを解決するものとして捉える。

本研究では、Hatori et al. (2012) にて用いられた、Huang と Sagae による動的計画法を用いたビーム探索の手法は使用していない (Huang and Sagae 2010; Hatori et al. 2012)。それは、Andor et al. (2016) によるニューラルネットワークのビームサーチを利用した広域最適化手法式が、遷移型解析の途中状態でのニューラルネットワーク関数の出力結果を入力とすることによる。遷移型解析にて経由するすべての途中状態でのニューラルネットワークの出力結果は、先の式を通じて、全体の計算グラフの中に取り込まれる。このように、ニューラルネットワークの計算グラフの中で、遷移の途中状態も含めた最適化が行われるために、少なくとも学習時は、この手法における動的計画法の採用は難しいと考えられる。また、ニューラルネットワーク固有の問題として Dropout を使用している場合は、そもそも、同一の入力に対して確率的に異なる振る舞いをする。この性質は、動的計画法には向かない。

### 3.4 双方向 LSTM を使用したモデル

3.3 節では、素性選択を利用したニューラルネットワークを提案した。このモデルは非常に高い性能を誇るが、以下の 2 つの問題が存在した。

- (1) ニューラルネットワークが限られた素性に基づいて動作するため、文全体の情報を入力にとることはできない
- (2) 素性選択に頼っている

この問題を解決するため、Kiperwasser and Goldberg (2016) は双方向 LSTM による構文解析モデルを提案した。彼らのモデルでは非常に少数の素性のみを使用しながら、双方向 LSTM を使用することで文全体の情報をニューラルネットワークに与えることに成功した。結果的に、彼らのモデルは、Penn Treebank データセットにて素性に基づく Weiss et al. (2015) によるモデルと比較可能な性能を達成した。彼らのモデルは、3 つの部分から成り立っている。入力文を処理する双方向

LSTM と、双方向 LSTM の隠れ層から素性を抽出する関数と、多層パーセプトロンである。語句を直接に素性として用いるのではなく、文全体を双方向 LSTM を利用して処理した各語句の分散表現を素性として用いることに特徴がある。

本論文では、Kiperwasser and Goldberg (2016) の手法を発展させ、双方向 LSTM による文全体の情報抽出を利用して、単純かつ大域的な素性を利用した統合構文解析モデルを提案する。Kiperwasser and Goldberg (2016) の手法は、入力が単語分割されていることを前提としている。したがって、中国語の統合構文解析に応用するに際し、単語分割が施されていない入力に、どのように双方向 LSTM を利用するかが問題となる。文字に対する単純な双方向 LSTM のみを使用した場合、文字列としての意味を捉えられるとは限らない。

そこで本研究では、単純な文字入力の他に、文中に存在するできる限り多くの文字列に対して、分散表現を付与し、その利用を可能にする方法を提案する。図 3 のように、入力文中に出現する複数の長さの文字列を捉えるために、 $n$  文字入力に対応する複数の双方向 LSTM を組み合わせる。 $n$  文字入力に対応する双方向 LSTM は文字列  $c_i \cdots c_{i+n-1}$  の配列である

$$\{c_1 \cdots c_n, \cdots, c_i \cdots c_{i+n-1}, \cdots, c_{N-n+1} \cdots c_N\}$$

を入力とする。ここで、 $c_i$  は  $N$  文字からなる文中の  $i$  番目の文字を表す。また、文終端記号やニューラルネットワークのパディングのための表現は別に与える。具体的には、 $n = 1$  のときは、文字入力の双方向 LSTM に対応し、 $n = 2$  のときは連続的な 2 文字  $c_i c_{i+1}$  を双方向 LSTM の入力とする。こうした  $n$  文字入力に対応する双方向 LSTM は、入力となる文字列の表現を、単語または文字の分散表現もしくは動的に生成される文字列の分散表現として利用する。このように文中に存在する文字列をできる限り利用する手法では、当然ながら、多くの不完全な単語が入力中に生成されてしまう。しかしながら、不完全な単語や文字列であっても、文字列の分散表現を利用することで、双方向 LSTM にその構成文字の情報を伝えることが可能となる。

このようにして生成された文中の文字列の双方向 LSTM による表現を、buffer 及び stack 内部の単語や文字列の表現として利用する。ただし、ここでは、文字  $c_i$  から開始する複数の文字列  $c_i, c_i c_{i+1}, \cdots, c_i \cdots c_{i+n-1}$  について、その双方向 LSTM による表現を連結したベクトル

$$[v(c_i), v(c_i c_{i+1}), \cdots, v(c_i \cdots c_{i+n-1})]$$

を、文字  $c_i$  から開始される文字列の素性の表現ベクトルとして利用する。双方向 LSTM を利用するモデルの素性は表 3 にまとめられる。最後に多層パーセプトロンおよび softmax 関数により、統合構文解析に用いられる遷移確率が予測される。

双方向 LSTM を使用するモデルについては、多層パーセプトロンには 3 層の重み行列を持つものを用いた。このニューラルネットワークについては、貪欲法による学習を適用した。

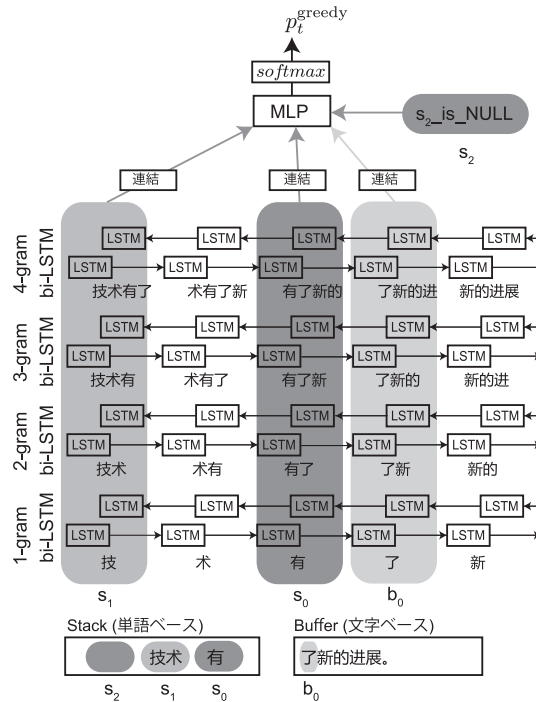


図 3 双方向 LSTM によるモデル.  $n$ -gram に対応する 4 個の独立した双方向 LSTM によって, 文全体が  $n$  文字ずつ処理され, 文字の開始位置に応じた 4 個の隠れ状態が作られる. 次に stack や buffer の状態に合わせて, 隠れ状態から素性抽出が行われる. なお, 素性抽出の際は, 単語の開始位置に対応する 4 個の隠れ状態を結合して用いる. なお, 本図においては, 4 素性モデルに対応する 4 個の素性を図示した. 8 素性モデルにおいて使用される, stack の子供の単語に関する 4 個の素性については, 図示していない.

表 3 双方向 LSTM を用いたモデルに使用される素性

モデル	素性
4 素性モデル	s0w, s1w, s2w, b0c
8 素性モデル	s0w, s1w, s2w, b0c s0r0w, s0l0w, s1r0w, s1l0w

構文解析を行う際の stack と buffer のみの非常に基本的な素性が 4 素性モデルであり, stack の子供の単語に関する 4 個の素性を加えたのが 8 素性モデルである. 素性に使用されている記号については, 表 2 と同一である.

### 3.5 係り受けラベルの推定モデル

本論文では係り受け解析のラベルの推定モデルを提案する. このモデルは, 先程までのモデルとは独立に, 入力文の単語列, 品詞タグおよびラベルなしの係り受け解析結果を入力にとる. モデル本体は, 入力単語列と品詞タグを処理する一層の双方向 LSTM と, 係り受けのある各単語ペアご

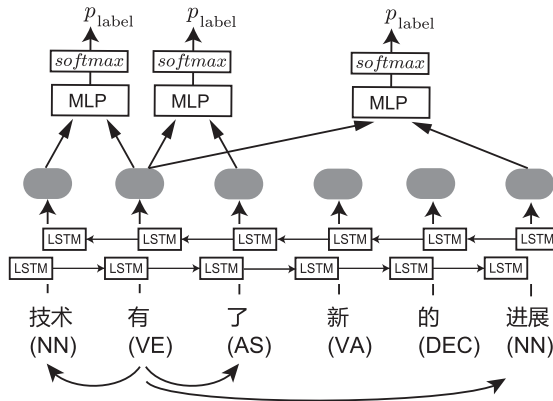


図 4 双方向 LSTM による係り受け解析結果へのラベル付けモデル. 係り受けのある単語対について, 係り受けラベルを推測する.

とにラベルを推定するフィードフォワードニューラルネットワークからなる (図 4).

前節までのモデルと同様に UNK に起因する問題を回避するため, このモデルは, 入力単語を, 単語, 文字および文字列の分散表現を利用して, 入力単語列の分散表現を得る. また, 品詞タグも分散表現に変換し, 入力単語列の分散表現と連結した上で, 双方向 LSTM への入力とする. 双方向 LSTM の出力から係り受け元と係り受け先の単語の分散表現を抽出し, 2 層フィードフォワードニューラルネットワークの入力とする. 2 層フィードフォワードニューラルネットワークの最終層にある softmax 関数が係り受けのラベルの予測分布を出力する.

## 4 実験

### 4.1 実験設定

実験には, 中国語の構文解析データセットである, Penn Chinese Treebank 5.1 (CTB-5) および Penn Chinese Treebank 7 (CTB-7) を利用した. データセットの分割には, 標準的に用いられている CTB-5 の分割 (Jiang, Huang, Liu, and Lü 2008) および CTB-7 の分割 (Wang, Kazama, Tsuruoka, Chen, Zhang, and Torisawa 2011) を使用した. オリジナルの CTB データセットには, 係り受け解析の結果は含まれていない. そこで, Penn2Malt<sup>2</sup>を使用して係り受け解析の形式に変換した. データセットの統計を表 4 に示す.

分散表現の事前学習には Chinese Gigaword Corpus を使用した. 分散表現の事前学習の具体的な手法は, 3.2 節の後半に記載した. 本研究では, 単語および文字の分散表現事前学習には word2vec を使用した (Mikolov, Chen, Corrado, and Dean 2013). 事前学習に用いたコーパスの単語分割に

<sup>2</sup> <https://stp.lingfil.uu.se/~nivre/research/Penn2Malt.html>



表 4 Penn Chinese Treebank データセットの統計

		#sentence	#oov
CTB-5	訓練データセット	18 k	—
	開発データセット	350	553
	テストデータセット	348	278
CTB-7	訓練データセット	31 k	—
	開発データセット	10 k	13 k
	テストデータセット	10 k	13 k

は, (Shen, Li, Choe, Chu, Kawahara, and Kurohashi 2016b) の KKN により行われた. これら単語と文字の同時埋め込み事前学習は, 3.2 節にて紹介した方法により行われた. 学習済み分散表現のうち, その頻度順の上位 100 万語を使用した. さらに, CTB データセットのうち訓練データセットに含まれる未知語については, 正規乱数を用いて初期化されたベクトルを使用する. 開発データセットおよびテストデータセットについては, このような事前に初期化された分散表現を持たない未知語を持つ.

本研究ではラベルなしの係り受け解析をまず実験し, 評価した. 次に, 4.2.7 節にてラベル付きの解析を行った. 単語分割や品詞タグ付け, 係り受けの評価には, Hatori et al. (2012) および Zhang et al. (2014) にしたがって, F1 測定による標準的な単語単位での評価を行った. この設定では, 正しく分割されていない単語に対しては, 品詞タグと係り受け解析の結果も正しいとはみなされない. Hatori et al. (2012) と同様に, 句読点への係り受けは評価の対象とはしない.

本研究では SegTag, SegTagDep および Dep の 3 種類のモデルを用いて実験を行った. SegTag は単語分割と品詞タグ付けの統合モデルである. SegTag モデルにて使用される遷移操作は 3.1 節にて, 使用される素性は, 3.3.2 節の末尾にて解説している. SegTagDep は単語分割・品詞タグ付けおよび係り受け解析の統合モデルであり, Dep は係り受け解析のモデルである. Dep モデルは Weiss et al. (2015) および Andor et al. (2016) に類似し, また使用される素性の種類は Chen and Manning (2014) と同様であるが, 素性の入力方法には, 3.2 節にて解説した文字列の分散表現を利用している. これにより, 単語分割の誤りや未知語などが Dep モデルの入力に含まれていても, Dep モデルは, 文字表現からある程度はその入力を捉えることが期待される. この効果は後ほど検証する.

この論文における実験の多くは GPU を用いて行われたが, ビームを用いた学習の一部は, 大きなミニバッチを使用していたために, GPU メモリに乗せることができず, CPU 上で行われた. ニューラルネットワークの実装には Theano を用いた.

## 4.2 結果

### 4.2.1 単語分割と品詞タグ付けの統合モデル

最初に、単語分割と品詞タグ付けの統合モデル (SegTag) について実験し、評価した。このモデルにて使用される遷移操作は 3.1 節にて、使用される素性は、3.3.2 節の末尾にて解説している。表 5 では、CTB-5 データセットを用いて、単語分割と品詞タグ付けの性能を比較した。本研究では提案手法を Hatori et al. (2012) および Zhang et al. (2014), Zhang, Li, Barzilay, and Darwish (2015) の 3 つの異なる関連研究と比較した。Zhang Y. らのモデルは Zhang M. らのモデルの k-best 出力を、再並び替えるモデルである (Zhang et al. 2015)。なお、この表における Hatori et al. (2012) の手法のスコアは Hatori et al. (2012) の論文から取得した。本研究での単語分割と品詞タグ付けの統合モデルは、これらの既存手法に対し、その両方で優れた性能を示した。この中には優れた辞書情報を利用する Hatori et al. (2012) のモデルも含まれる。ただし、本研究では、Chinese Gigaword Corpus を用いて、単語および文字の分散表現を事前学習しており、CTB データセットのみを使用している既存手法との比較においては注意を要する。

### 4.2.2 単語分割・品詞タグ付けおよび係り受け解析の統合モデル

表 6 は単語分割・品詞タグ付けおよび係り受け解析までをすべて統合して行った場合の結果を示している。特に貪欲法を用いて学習を行った統合モデルである SegTagDep(g) モデルとビームを用いて学習を行った SegTagDep モデルの結果を示す。提案手法は単語分割と品詞タグ付けにおいて既存手法を超える性能を発揮した。Andor et al. (2016) によるビームサーチを用いることで係り受け解析のスコアは更に改善されたが、Zhang M. らの手法およびそれを用いる Zhang Y. らの手法にはわずかに劣る性能を示した。なお、この表における Hatori et al. (2012) の手法のスコアは

表 5 単語分割と品詞タグ付けの統合モデル

モデル	単語分割	品詞タグ付け
Hatori+12 SegTag	97.66	93.61
Hatori+12 SegTag(d)	98.18	94.08
Hatori+12 SegTagDep	97.73	94.46
Hatori+12 SegTagDep(d)	98.26	94.64
M. Zhang+14 EAG	97.76	94.36
Y. Zhang+15	98.04	94.47
SegTag(g)	98.41	<b>94.84</b>
SegTag	<b>98.60</b>	94.76

CTB-5 における実験結果。SegTag モデルを、Hatori et al. (2012) および Zhang et al. (2014), Zhang et al. (2015) のモデルと比較した。Hatori et al. (2012) のモデルにおける (d) は、辞書を使用したモデルであることを示す。Zhang et al. (2014) のモデルにおける EAG は Arc-Eager モデルであることを示す。SegTag(g) は貪欲法を用いて学習されたモデルを、SegTag はビームサーチを用いて学習されたモデルを示す。

表 6 単語分割・品詞タグ付けおよび係り受け解析の統合モデル (SegTagDep) と単語分割と品詞タグ付けの統合モデルと係り受け解析のパイプラインモデル (SegTag+Dep).

モデル	単語分割	品詞タグ付け	係り受け (UAS)
Hatori+12	97.75	94.33	81.56
M. Zhang+14 STD	97.67	94.28	81.63
M. Zhang+14 EAG	97.76	94.36	81.70
Y. Zhang+15	98.04	94.47	82.01
SegTagDep(g)	98.24	94.49	80.15
SegTagDep	98.37	<b>94.83<sup>‡</sup></b>	81.42 <sup>‡</sup>
SegTag+Dep	<b>98.60<sup>‡</sup></b>	94.76 <sup>‡</sup>	<b>82.60<sup>‡</sup></b>

CTB-5 における実験結果. (g) は貪欲法を用いて学習されたことを示す. SegTag+Dep モデルにおいては, SegTag と Dep の双方でビームを用いて訓練とテストを行っている. なお, Zhang et al. (2015) は他の構文解析器の結果を再配置し, 精度の向上を目指すモデルである. † は SegTagDep(g) モデルに対して対応のある t 検定を行い,  $p < 0.01$  で統計的に有意な改善が見られたことを示す.

Zhang et al. (2014) から取得した.

なお, 貪欲法によるモデルである SegTagDep(g) はビームサーチを用いるモデルよりわずかに劣る結果となったが, ビームを用いないことによる解析の速さは一考に値する.

#### 4.2.3 単語分割と品詞タグ付けの統合モデルと係り受け解析のパイプラインモデル

次に SegTag モデルと依存構造解析のみのモデル (Dep) とのパイプラインモデル (SegTag+Dep) を実験した. 簡単のため, SegTag モデルは貪欲法で学習されたのに対し, Dep モデルはサイズ 4 のビームを用いて学習及びテストが行われた. 結果を表 6 の SegTag+Dep に示す. 同じくビームサーチを使用するモデルである SegTagDep と SegTag+Dep を比較した場合, SegTag+Dep は係り受け解析を含めた統合モデルである SegTagDep よりも係り受け解析および単語分割において優れていた.

SegTagDep モデルや SegTag+Dep モデルにおける単語分割の誤りは, 主に固有表現にて生じている. 本論文では, これを詳細に考察する. SegTagDep モデルにて用いられるビームを用いた学習では, Hatori et al. (2012) の AP 操作に 2 倍の配置長を持たせる手法が使われている. これは, 単語内部の文字同士の係り受けを考慮していると解釈できる. 例えば“記者”(記者), “实验室”(実験室) のような単語においては, 前の文字が, 順番に後ろの文字に係り受け構造を持つとして, 直感的にも理解できる. この解釈では, AP 操作は, 文字を連結し, かつ単語内の係り受けを解決しているために, 2 倍の配置長を持つ操作となる.

しかし, 固有表現においては, 先程のような単語内係り受けが存在するとは考えにくい場合がある. 例えば, 人名である“卢仁法”は文字同士の係り受けを持たない単語と見なしたほうが自然である. そのような単語に対しては, SegTagDep モデルでは SH 遷移を行うことが多くなる. これは

Hatori et al. (2012) の AP 遷移が文字の結合と単語内係り受けの解決という 2 つのことをしているとみなした場合に、固有表現には必ずしもそのような内部の係り受けが存在するとは限らないため、やや過剰な配置をもたせていると考えられるためである。そのため、結果的に、SegTag+Depの方がSegTagDepより優れた単語分割結果となったと考えられる。

固有表現について、SegTagDepモデルとSegTag+Depモデルを比較した場合、以下のような差が見られた。“总统 龚保雷 餐叙。”(ゴンバオリ大統領が会食をして、)という句中の“龚保雷 餐叙”に対し、SegTagDepモデルは“龚保雷餐叙”という一単語として固有表現としたのに対し、SegTag+Depモデルは固有表現の“龚保雷”(ゴンバオリ・人名)と動詞の“餐叙”(会食をして)として、正しく単語分割を行った。地名についても、一語の固有表現である“新喀里多尼亚”(ニューカレドニア)や“所罗门群岛”(ソロモン諸島)に対して、SegTagDepモデルは過分割を行ったが、SegTag+Depモデルは一語の固有表現として、正しく分割を行った。やや特殊な例としては、“以多明尼加为例, 就在陈水扁总统造访前夕, 多国新总统阵营主动向媒体透露, 中国大陆将为多国兴建大型火力发电厂的讯息。”(ドミニカ共和国では、陳水扁総統が訪れる直前に、新大統領陣営は自ら中国大陸により大型の火力発電所が建設されることをメディアに告げた。)という文では、“多国”がドミニカを指し示す固有表現となる。一方で、“多国”は“多”および“国”と二単語にも分割され、この場合は「多くの国」という意味にもなる。このような場合に、SegTagDepでは“多国”を“多”と“国”に分割して出力されたが、SegTag+Depでは「多国」として一語の固有表現として出力された。

また、関連研究にて、Zhang et al. (2014) の提案した STD (Arc-standard) モデルは、Hatori et al. (2012) の単語分割・品詞タグ付けおよび係り受け解析の統合モデルよりも高性能に動作した。Zhang et al. (2014) の提案した STD (Arc-standard) モデルは、まず単語分割を先に行い、その結果を“deque”に格納してから係り受け解析を行うモデルであり、本論文におけるSegTag+Depモデルと類似する。

本研究における提案手法においては、加えて、単語と文字の分散表現を利用することで従来法より高性能に単語分割と品詞タグ付けが動作するために、単語分割と品詞タグ付けの結果を先読みすることでDepモデルが最終的に良い結果を出しているものと推測される。

#### 4.2.4 CTB-7 における実験

SegTagDepおよびSegTag+Depモデルをより大きなCTB-7データセットにおいても評価した。CTB-7データセットにおける実験においては、4層の隠れ層を持つ多層パーセプトロンが3層のものよりも優れた結果を残した。ただし、CTB-5における実験においては、明確な差異を見出すことができなかった。この違いについては、訓練データセットのサイズの違いによるものと推測される。4層の隠れ層を持つ多層パーセプトロンモデルによる実験結果を表7に示す。

#### 4.2.5 文字列の分散表現の影響

本研究にて提案する文字列の分散表現がどの程度有効なのかを調べるため、SegTag+Dep モデルについて、文字列の分散表現を使用した場合と UNK に対応する分散表現を使用した場合の、性能の変化を調べた。実際の使用条件に近い状況で試験を行うために、SegTag モデルによる単語分割及び品詞タグ付けの解析結果に対し、係り受け解析を行う。係り受け解析には、貪欲法により学習された以下の2つのモデルを試験した。最初のモデル Dep(g) は、文字列の分散表現を使用した係り受けモデルである。次のモデル Dep(g)-cs は、文字列の分散表現を使用せず、文字と単語の分散表現に出現しない未知語に対しては、UNK の分散表現を使用したモデルである。入力素性に対応する分散表現が存在しなかった場合には、従来法である Dep(g)-cs モデルは UNK の分散表現を使用するが、Dep(g) は文字列の分散表現を使用する。結果を表 8 に示す。なお、簡単のために、ビームを使用しない条件での比較を示す。

この結果から、モデルが未知語に遭遇した場合、単純な UNK の分散表現を利用するよりも、文字列の分散表現を利用したほうが性能が良くなることがわかる。

表 7 SegTagDep と SegTag+Dep の実験結果

モデル	単語分割	品詞タグ付け	係り受け (UAS)
Hatori+12	95.42	90.62	73.58
M. Zhang+14 STD	95.53	90.75	<b>75.63</b>
SegTagDep(g)	96.06	90.28	73.98
SegTagDep	95.86	<b>90.91</b>	74.04
SegTag+Dep	<b>96.18</b>	90.90	75.12

CTB-7 における実験結果。なお、貪欲法を用いて訓練されたモデル SegTagDep(g) による結果も同様に示す。SegTag+Dep モデルにおいては、SegTag と Dep の双方でビームを用いて訓練とテストを行っている。

表 8 SegTag の解析結果に対し、文字列の分散表現を利用した場合と UNK の分散表現を利用した場合の Dep(g) モデルの性能

モデル	係り受け (UAS)
Dep(g)-cs	80.51
Dep(g)	80.98*

CTB-5 における実験結果。“Dep(g)-cs” は、文字列の分散表現を使用せず、文字と単語の分散表現に出現しない未知語に対しては、UNK の分散表現を使用したモデルである。\* は対応のある t 検定を行ったところ、 $p < 0.05$  で統計的に有意な改善が見られたことを示す。

#### 4.2.6 双方向 LSTM を使用したモデル

双方向 LSTM を使用したモデルについても、同様に実験を行った。なお、本実験では貪欲法に基づく学習のみを行った<sup>3</sup>。双方向 LSTM を利用したモデルは、表 3 のように基本的な素性にしか依存しないという利点がある。結果を表 9 に示す。双方向 LSTM を用いたモデルは、従来法と比較してやや悪い性能となったが、非常に基本的な素性のみを使用してこの性能を達成したことは注目に値する。

#### 4.2.7 係り受けのラベル推定

最後に、係り受けのラベル解析モデルを利用することで、ラベルも含めた係り受け解析を行う。CTB-5 および CTB-7 に対して、係り受けのラベルの推定を行った。この実験では、ラベルなしの評価では最も優れていた SegTag+Dep の解析結果に対し、更にラベル推定を行うことで、各コーパスにおける原文からラベル付き係り受け解析を行った際のスコアを測定することを目的とする。係り受け解析のラベルの推定結果を表 10 に示す。この係り受けラベルの推定モデルについては、パイプラインもしくは統合モデルによる評価が、Hatori et al. (2012) や Zhang et al. (2014) などの先行研究では行われておらず、比較の対象とはしなかった。

また、上の CTB-5 と CTB-7 の解析結果について、ラベル付き係り受けのうち、もととなるラベルなしの係り受けが正しい係り受けについてのラベル毎の正解率を集計した。その結果を表 11

表 9 双方向 LSTM を使用するモデル

モデル	単語分割	品詞タグ付け	係り受け (UAS)
Hatori+12	97.75	94.33	81.56
M. Zhang+14 EAG	97.76	94.36	81.70
SegTagDep(g)	98.24	94.49	80.15
双方向 LSTM 4 素性モデル (g)	97.72	93.12	79.03
双方向 LSTM 8 素性モデル (g)	97.70	93.37	79.38

CTB-5 における実験結果。4 素性モデルと 8 素性モデルは、表 3 における使用素性数と対応する。

表 10 ラベルなし係り受け解析スコア (UAS) とラベル付き係り受け解析スコア (LAS)

モデル	CTB-5	CTB-7
SegTag+Dep (UAS)	82.6	75.1
SegTag+Dep (LAS)	80.0	72.3

CTB-5 と CTB-7 における実験結果。LAS は、UAS と同様に単語分割が正しい単語同士の係り受けの場合のみ正解となる。また、引用符や句読点は評価から取り除かれている。

<sup>3</sup> ビームを用いた学習により性能の向上が見込まれるが、双方向 LSTM を使用していることにより、現実的な時間内での訓練及び評価は難しいと判断した。

表 11 ラベルごとの正解率

	PRD	VC	SUB	OBJ	VMOD	DEP	NMOD	PMOD	AMOD	SBAR
CTB-5	87.4	91.7	94.3	95.7	94.6	94.5	98.5	97.5	95.9	90.7
CTB-7	93.6	84.8	90.6	94.2	97.0	97.0	98.1	97.6	99.1	96.2

CTB-5 と CTB-7 における実験結果. なお, 句読点の係り受けである “P” ラベルと ROOT への係り受けである “ROOT” ラベルは除いた.

に示す. 総じて, 修飾語 (modifier) はスコアが高めとなるが, 文内のより大域的な構造である主格 (SUB) や目的格 (OBJ) については低めとなる傾向が見られた.

## 5 将来研究

本研究中では, 結果として, 単語分割, 品詞タグ付けおよび係り受け解析の統合モデルよりも, 単語分割と品詞タグ付けの統合モデルおよび係り受け解析のパイプラインモデルの方が, 優れた性能を発揮した. これは, 係り受け解析の情報を単語分割に活用する際に, どこまで内部的に単語分割を行ってから, 係り受けを解析を行うか, という問題と関連する. 例えば, Hatori et al. (2012) のモデルでは, 単語分割と係り受け解析は同時であると言えるが, Zhang et al. (2014) の STD モデルでは, 単語分割を 3 単語分だけ先に行うことにより, 係り受け解析の際に, 単語の先読みを可能にしている. このような問題に対し, 解析文やモデルの状態に応じて, 単語の先読みを行うか, 係り受け解析を優先して行うかを判断するモデルを作成することができれば, 非常に有用な統合構文解析器となるだろう.

この他, 本論文では, 最後に係り受けのラベル推定を行ったが, この係り受けラベル推定モデルも含めて係り受け解析を行うモデルも考えられる. このようにすることで, 係り受けのラベルの種類に関する情報が係り受け解析や, 品詞タグ付けなどの精度に良い影響を及ぼしうるため, これは有望な研究となりえるだろう.

最後に, 双方向 LSTM による文中の単語表現は, 例えば, 近年提案された ELMo のように, 非常に柔軟かつ強力な手法である (Peters, Neumann, Iyyer, Gardner, Clark, Lee, and Zettlemoyer 2018). しかしながら, 今回の研究で速度および GPU メモリの制約により, 双方向 LSTM を用いるモデルとビーム探索を併用することが出来なかったように, 中国語の統合解析への双方向 LSTM の使用は, あまり容易なことではない. これは, 主に, 中国語の統合構文解析では, 解析が進むに連れて, 文字ばかりでなく解析済みの単語の情報を利用することが重要となるからである. 解析のステップごとに, 解析済みの単語を用いて双方向 LSTM を再計算する手法は, 計算コストがかなり大きいと言える. この問題点が解決されれば, この手法を用いて世界最高性能を達成するモデルを作成しうると考えられる.

## 6 結論

本論文では、中国語の統合構文解析を行う2つのモデルを提案した。1つ目は素性とフィードフォワードニューラルネットワークに基づくモデルであり、2つ目は双方向 LSTM を使用したニューラルネットワークに基づくモデルである。そのいずれにおいても、文字列の分散表現を使用することで、UNK に相当する分散表現を用いることを避け、不完全な語句であっても、その構成する文字の分散表現から文字列同士の分散表現を得ることが可能となる。結果として、単語分割、品詞タグ付けおよび係り受け解析の統合モデルは、中国の単語分割と品詞タグ付けにおいて既存の統合解析手法よりも優れた性能を発揮した。しかし、係り受け解析については、既存手法と比較可能、もしくはやや劣る性能にとどまった。そこで、さらに、単語分割と品詞タグ付けの統合モデルおよび係り受け解析のパイプラインモデルを実証することで、係り受け解析にても世界最高の性能を達成した。双方向 LSTM モデルを使用することで、基本的な素性のみを使用しながら文全体の構造も考慮するモデルも提案した。

## 謝 辞

この論文は ACL2017 にて発表を行った “Neural Joint Model for Transition-based Chinese Syntactic Analysis” を和訳し、拡張したものです。

## 参考文献

- Alberti, C., Weiss, D., Coppola, G., and Petrov, S. (2015). “Improved Transition-Based Parsing and Tagging with Neural Networks.” In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1354–1359. Association for Computational Linguistics.
- Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., and Collins, M. (2016). “Globally Normalized Transition-Based Neural Networks.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2442–2452. Association for Computational Linguistics.
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). “Neural Machine Translation by Jointly Learning to Align and Translate.” *CoRR*, **abs/1409.0473**.
- Ballesteros, M., Dyer, C., and Smith, N. A. (2015). “Improved Transition-based Parsing by Modeling Characters instead of Words with LSTMs.” In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 349–359. Association for



Computational Linguistics.

- Chen, D. and Manning, C. (2014). “A Fast and Accurate Dependency Parser using Neural Networks.” In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 740–750. Association for Computational Linguistics.
- Cross, J. and Huang, L. (2016). “Incremental Parsing with Minimal Features Using Bi-Directional LSTM.” In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 32–37, Berlin, Germany. Association for Computational Linguistics.
- Duchi, J., Hazan, E., and Singer, Y. (2010). “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.” No. UCB/EECS-2010-24.
- Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). “Transition-Based Dependency Parsing with Stack Long Short-Term Memory.” In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 334–343. Association for Computational Linguistics.
- Hatori, J., Matsuzaki, T., Miyao, Y., and Tsujii, J. (2011). “Incremental Joint POS Tagging and Dependency Parsing in Chinese.” In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 1216–1224. Asian Federation of Natural Language Processing.
- Hatori, J., Matsuzaki, T., Miyao, Y., and Tsujii, J. (2012). “Incremental Joint Approach to Word Segmentation, POS Tagging, and Dependency Parsing in Chinese.” In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1045–1053. Association for Computational Linguistics.
- Hinton, G. E., McClelland, J. L., and Rumelhart, D. E. (1986). “Learning Distributed Representations of Concepts.” In *Proceedings of the 8th Annual Conference of the Cognitive Science Society*. Vol. 1, p. 12.
- Huang, L. and Sagae, K. (2010). “Dynamic Programming for Linear-Time Incremental Parsing.” In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1077–1086, Uppsala, Sweden. Association for Computational Linguistics.
- Jiang, W., Huang, L., Liu, Q., and Lü, Y. (2008). “A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging.” In *Proceedings of ACL-08: HLT*, pp. 897–904. Association for Computational Linguistics.
- Kingma, D. P. and Ba, J. (2015). “Adam: A Method for Stochastic Optimization.” In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th*

- International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.
- Kiperwasser, E. and Goldberg, Y. (2016). “Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations.” *Transactions of the Association for Computational Linguistics*, **4**, pp. 313–327.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). “Efficient Estimation of Word Representations in Vector Space.”
- Nair, V. and Hinton, G. E. (2010). “Rectified Linear Units Improve Restricted Boltzmann Machines.” In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, June 21–24, 2010, Haifa, Israel, pp. 807–814.
- Nivre, J. (2004). “Incrementality in Deterministic Dependency Parsing.” In Keller, F., Clark, S., Crocker, M., and Steedman, M. (Eds.), *Proceedings of the ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together*, pp. 50–57. Association for Computational Linguistics.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). “Deep Contextualized Word Representations.” In *Proceedings of NAACL*.
- Shen, M., Kawahara, D., and Kurohashi, S. (2016a). “Chinese Word Segmentation and Unknown Word Extraction by Mining Maximized Substring.” 自然言語処理, **23** (3), pp. 235–266.
- Shen, M., Li, W., Choe, H., Chu, C., Kawahara, D., and Kurohashi, S. (2016b). “Consistent Word Segmentation, Part-of-Speech Tagging and Dependency Labelling Annotation for Chinese Language.” In *Proceedings of the 26th International Conference on Computational Linguistics (COLING2016)*, pp. 298–308, Osaka, Japan.
- Wang, Y., Kazama, J., Tsuruoka, Y., Chen, W., Zhang, Y., and Torisawa, K. (2011). “Improving Chinese Word Segmentation and POS Tagging with Semi-supervised Methods Using Large Auto-Analyzed Data.” In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 309–317, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.
- Weiss, D., Alberti, C., Collins, M., and Petrov, S. (2015). “Structured Training for Neural Network Transition-Based Parsing.” In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 323–333. Association for Computational Linguistics.
- Zhang, M., Zhang, Y., Che, W., and Liu, T. (2014). “Character-Level Chinese Dependency Parsing.” In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1326–1336. Association for Computational Lin-

guistics.

- Zhang, Y., Li, C., Barzilay, R., and Darwish, K. (2015). “Randomized Greedy Inference for Joint Segmentation, POS Tagging and Dependency Parsing.” In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 42–52. Association for Computational Linguistics.
- Zhang, Y. and Clark, S. (2008). “A Tale of Two Parsers: Investigating and Combining Graph-based and Transition-based Dependency Parsing.” In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 562–571. Association for Computational Linguistics.
- Zhang, Y. and Clark, S. (2010). “A Fast Decoder for Joint Word Segmentation and POS-Tagging Using a Single Discriminative Model.” In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 843–852. Association for Computational Linguistics.
- Zhou, H., Zhang, Y., Huang, S., and Chen, J. (2015). “A Neural Probabilistic Structured-Prediction Model for Transition-Based Dependency Parsing.” In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1213–1222. Association for Computational Linguistics.

## 略歴

**栗田 修平**：2013年京都大学理学部物理系卒業。2015年同大学院物理学教室修士課程修了。2015年より京都大学大学院情報学研究科知能情報学専攻博士課程在籍中。2019年同大学院博士取得見込み。深層学習，深層学習を用いた自然言語処理の研究に従事。修士（理学）。言語処理学会，ACL，各会員。

**河原 大輔**：1997年京都大学工学部電気工学第二学科卒業。1999年同大学院修士課程修了。2002年同大学院博士課程単位取得認定退学。東京大学大学院情報理工学系研究科学術研究支援員，独立行政法人情報通信研究機構研究員，同主任研究員を経て，2010年より京都大学大学院情報学研究科准教授。自然言語処理，知識処理の研究に従事。博士（情報学）。情報処理学会，言語処理学会，人工知能学会，電子情報通信学会，ACL，各会員。

**黒橋 禎夫**：1994年京都大学大学院工学研究科電気工学第二専攻博士課程修了。博士（工学）。2006年4月より京都大学大学院情報学研究科教授。自然言語処理，知識情報処理の研究に従事。言語処理学会10周年記念論文賞，同20周年記念論文賞，第8回船井情報科学振興賞，2009 IBM Faculty Award 等を受賞。2014

年より日本学会議連携会員.

(2018年8月1日 受付)

(2018年11月1日 再受付)

(2018年12月13日 採録)