直接分類型日本語ニューラル構文解析

金山 博　　村岡 雅康　　小比田涼介
{hkana,mmuraoka}@jp.ibm.com　　kohita.ryosuke.kj9@is.naist.jp
日本アイ・ビー・エム株式会社 東京基礎研究所　　奈良先端科学技術大学院大学

1　はじめに

近年の構文解析器はニューラルネット (NN) による実装が標準的となっており、2017 年に行われた 49 言語の構文解析のシェアドタスク [12] では、上位のシステムのほとんどが (bi-)LSTM や分散表現など NN の手法を利用していた [4, 11]。NN による構文解析器の多くは遷移型 [9] やグラフ型 [8] のアルゴリズム、またはそれらの組み合わせ [10] をベースにしており、文内の双方向の依存構造や交差がある場合など、多言語の分析における諸問題を解決している。一方で、遷移型の場合は特に、学習の対象であるアクション列が直感的に理解しづらい、最終的な結果として（スコアは高いにしても）直感的でない構造が出力されうる、といった問題がある。グラフ型の場合も文全体の大域的な構造を捉えられるものの、他のノードとの整合が取りづらい場合がある。

日本語の構文解析においては、先述のシェアドタスクでは高い精度が出ておらず[*1]、他の言語と共通の方法が有効に適用できているとはいえない。内容語を主辞とした単語単位の係り受けの構造では、他の言語と同様に係り先が双方向になるが、文節単位の係り受けに注目して、倒置や注釈など特殊な場合を除外すれば、日本語の主辞後置性を活かした手法を適用することができる。本稿では、限定された少数の候補から係り先を選択する分類問題 [6] に対して NN の手法を適用して、文法知識と統合しながら制御可能で直感的な出力が得られる構文解析器の構築を目指す。

2　3 つ組/4 つ組モデル

日本語係り受け解析のための 3 つ組/4 つ組モデル [6, 15] とは、文法及びヒューリスティクスを用いて係り先候補を高々 3 つに絞り、係り元文節とすべての係り先候補文節の属性を用いて、それぞれの候補に係る確率を求める手法である。以下にその概要を示す。

[*1] なお、数値の低さの原因は、少量の学習データで単語区切りをする困難さによるものが大きい。

表 1　簡素化した日本語の文法規則の例。

係り属性の定義
・語形が格助詞「の」: 連用・連体
・語形が格助詞「を」: 連用
・語形が副詞: 連用・副詞
受け属性の定義
・名詞を含む文節: 連体
・用言・判定詞を含む文節: 連用
・副詞を含む文節: 副詞
修飾可能性の追加
・語形が格助詞「と」→ 「一緒に」の文節
・語形が格助詞「を」 →主辞が「条件 | 中心 | きっかけ | … ＋ に」の文節
・文節が「全部で」→ 数詞を含む文節

表 2　係り先候補数に対する、正しい係り先の分布（単位は %）。「比率」は、候補数の分布を示す。括弧付きの値は他の項との重複を表す。★は、第一・第二・最遠のいずれかに係る割合である。

候補数	比率	第一	第二	第三	第四	..	最遠	★
1	32.7	100	-	-	-	-	(100)	100
2	28.1	74.3	26.7	-	-	-	(26.7)	100
3	17.5	70.6	12.6	(16.8)	-	-	16.8	100
4	9.9	70.4	11.1	4.7	(13.8)	-	13.8	95.3
5	5.4	70.1	11.6	4.2	2.5	..	11.5	93.2
>5	6.4	70.3	10.8	3.9	2.4	..	9.6	90.7
計	100	-	-	-	-	..	-	98.6

2.1　係り先候補の絞り込み

まず、各文節が、同一文内でその文節より右側に位置する文節を修飾し得るか否かを、文法を用いて決定する。その際には、表 1 に示すような文法規則を考え、係り元文節の係り属性と同じ受け属性を持つ文節を、修飾可能な文節として列挙する [16, 13]。なお、主辞とは文節内の最右の自立語、語形とは文節内で読点を除く最右の形態素である。

表 2 に示す通り、修飾可能であるとされた文節集合のうち、係り元文節から最も近い文節、2 番目に近い文節、最も遠い文節に係る場合が 98.6 %を占める。このことから、修飾可能な文節が 4 つ以上ある場合も上記の 3 文節のみを考慮し、他の文節は無視することにす

る。以降では、このように 3 つ以下に制限された文節集合を、単に係り先候補と呼ぶ。

2.2 係り受けの計算とモデルの特徴

文節 b が、左から n 番目の係り先候補文節 c_{bn} に係る確率 $P(b \to c_{bn})$ を、文節 b の属性 Φ_b 及び c_{bn} の属性 $\Psi_{c_{bn}}$[*2]を用いて、係り先候補が 2 つの場合は 3 つ組の式 (1)、係り先候補が 3 つの場合は 4 つ組の式 (2) で計算する。

$$P(b \to c_{bn}) = P(n \mid \Phi_b, \Psi_{c_{b1}}, \Psi_{c_{b2}}) \quad (1)$$
$$P(b \to c_{bn}) = P(n \mid \Phi_b, \Psi_{c_{b1}}, \Psi_{c_{b2}}, \Psi_{c_{b3}}) \quad (2)$$

文全体の構造の確率 $P(T)$ は、上記の係り確率が独立であるという仮定に基づいて、文中の各係り受け確率の積

$$P(T) \simeq \prod_b P(b \to c_{bn}) \quad (3)$$

として[*3]、$P(T)$ が最大となるような係り受けを後方からのビームサーチにより探索する。

式 (1), (2) の特徴は、「係り元文節とその係り先候補の全ての属性を同時に考慮する」こと、そして「それぞれの係り先候補への係りやすさを求めるのではなく、各候補が選ばれる確率を直接求める」ことである。これにより、係り元文節から見た各候補の相対的な位置、他の候補の属性との相互関係（文脈情報）などを自然に反映させることができる。

3 NN モデルの設計

従来の 3 つ組/4 つ組モデルによる構文解析 [6, 13] では 1, 2 の式を、多数の二値素性を用いて各文節の属性を表現したうえで、ロジスティック回帰 (最大エントロピー法) で計算していた。今回はこれを、Chen らによる手法 [2] と同様に、単語や品詞の分散表現を用いた NN に置き換える。本研究で用いる手法である、係り先を直接選択するニューラルネットを図 1 に示す。ここでは例として、「…アユを、1953 年に絶滅した A 県の B 地区に提供、アユを蘇らせる計画が進行中だ。」という文の下線部の「アユを」の係り先を、3 つの係り先候補から選択する状況であり、正解は候補 2 である。

まず、それぞれの文節の主辞と、語形の 2 単語の表層と品詞を分散表現に変換する。すなわち、3 つ組モデル（係り先候補が 2 つの場合）では 6 単語、4 つ組モデル（候補が 3 つの場合）では 8 単語に対して 2 つずつの分散表現を用いる。さらに、係り元文節と係り先候補の 2 文節間の属性として以下の情報をベクトル化したものを用いる。

- 文節間に含まれる助詞「は」の数 (1, 2, 3, 4, 5 以上)
- 文節間に含まれる読点の数 (1, 2, 3, 4, 5 以上)
- 文節間の距離 (1, 2, …, 9, 10 以上)

これらを単一の隠れ層に伝え、出力として得られる 1, 2, 3 の高々 3 つの値に対する確率を softmax にて求める。

係り元文節に対して各候補に係る確率が求まるが、これらは係り元ごとに独立に求めているため、文全体では係り受けの交差が起こり得る。そこで、右側の文節から順に式 (3) の値をもとにビームサーチを行うとともに、交差を伴う係り方のペアを排除するようにして、文全体の構文木を構成する。

表 3　文節係り受けの精度。

学習手法	学習サイズ	係り受け正解率	
logistic 回帰	19k	88.92%	(19468/21894)
NN	4k	88.15%	(19300/21894)
NN	8k	88.49%	(19373/21894)
NN	12k	89.17%	(19522/21894)
NN	16k	**89.31**%	(19554/21894)

4 実験

4.1 データと設定

EDR 日本語コーパス [5] の約 20 万文のうち、文節の単位が解析結果と一致する文を抽出し、160,080 文を学習用、8,829 文を開発用、2,941 文をテストに用いた。学習には TensorFlow [1] を用い、損失関数はクロスエントロピーで計算し、L2 正則化項として 10^{-8} を乗じたものを加え、AdamOptimizer [7] で最適化した。単語は表層形を 100 次元のベクトル、74 種の品詞細分類に読点の有無を加えた 148 種を 50 次元のベクトル、その他の文節間の素性はそれぞれを 10 次元のベクトルとして、ランダムに初期化したものを学習時に更新した。すなわち、3 つ組モデルでは 990 次元、4 つ組モデルでは 1,320 次元のベクトルで埋め込み層を表現する。隠れ層は 200 次元、ビーム幅は 5 に設定した。

4.2 実験結果

実験の結果を表 3 に示す。文法による候補の絞り込みと素性のデザインを行い、約 19 万文のコーパスを用

[*2] $\Psi_{c_{bn}}$ には係り元文節と係り先候補の間の文節の属性も含まれているため、厳密には c_{bn} だけでなく b にも依存する。
[*3] 係り受けが交差しない制約を加えるため、この式は厳密ではない。

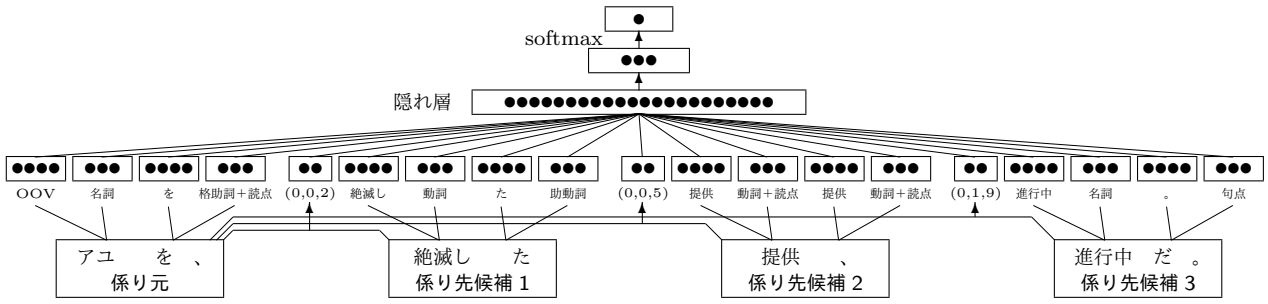

図 1　4 つ組モデルで 3 つの候補から係り先を選択するニューラルネット。係り元を含む 4 つの文節の主辞と語形、そして文節間の属性が用いられている。

表 4　内容語語彙・読点の属性の有無と精度。

内容語語彙	読点	係り受け正解率
○	○	89.31%
×	○	88.95%
○	×	87.40%
×	×	87.24%

いて logistic 回帰で学習した時のもの [13] と比較したところ、同様の文法を用いた後に NN にて学習した場合にはコーパスが 12 万文を超えたところから精度が上回り、16 万文を用いた学習では従来よりも 0.4 ポイントの向上が見られた。

今回は係り元と係り先の候補の文節にある語の情報だけを用いており、その他の文脈や、他の文節の係り受けについては考慮していない。文法で係り先を絞り込むことによって、初期の遷移型ニューラル構文解析 [2] では最大 18 単語を見ていたのに比べて、少ない情報で係り受けを解決できている。

なお、学習にかかる時間には大きな差があった。従来は CPU で各モデルの学習に 2〜8 時間を要していたが、NN の学習は GPU 上で 5〜15 秒で収束に至った。

文節の情報のうち性能に寄与したものを見るために、一部の属性を外して学習する ablation を行った。特に、分散表現を用いることによって従来手法よりも多くの内容語の語彙を扱えることによる効能と、日本語の係り受け解析において大きな意味を持つ読点 [14] の寄与について調べた。表 4 の結果を見ると、内容語語彙（語彙の大きさは 11,362）の寄与は驚くほどに小さく、それらを全て無視した場合にも精度の低下は 0.36 ポイントに留まった。一方で、読点の存在を無視した（品詞に読点の有無を組み入れず、かつ文節間の読点の数の素性を外した）場合には 2 ポイント近くの精度の低下が見られ、読点の重要性が確認できる。

図 1 に示した係り先の推定は、今回の NN の手法で正しく解析できた例である。従来の手法や、語彙を外した時には正しく解析されない（「絶滅した」に係ってしまう）ので、格助詞「を」と「提供」の間の親和性が距離の遠さに打ち勝つことが学習されていると考えられる。

4.3　その他の試行

その他、以下のような拡張を試したが、精度の向上は見られなかった。これらの点については今回の実験の設定が妥当であったといえる。

- 係り元文節の前後にある文節の情報の組み入れ。特に左側の文節は現状ではまったく考慮されていないが、加えても精度の増加は見られなかった。
- 学習済みの単語の分散表現の利用。そもそも内容語語彙の寄与が大きくないことと、構文的な属性と意味の間に差があることから、ランダムに初期化したものを学習時に更新するだけで充分であった。
- 文字の素性の導入。内容語の表層形の末尾 2 文字などを導入したが、係り先の決定には寄与しなかった。
- 単語や品詞の埋め込みベクトルや隠れ層の次元の拡大・縮小。
- 4 つの係り先候補を同時に考慮する 5 つ組モデル。従来では候補の数が増えた場合の学習が困難であったのに対し、今回は充分に短い時間で学習ができたが、例えば係り先候補が 4 つある時に 3 番目の候補が正しい係り先となる場合は全体の 0.5% に満たないなど、3 つ組・4 つ組でカバーできる範囲で充分であった。

5　おわりに

文法規則によって係り先候補を絞り込んで係り先を選択する手法を、ニューラルネット上での実装と統合することができ、かつ従来の機械学習よりも高い精度が得られることが確認できた。多くの NN の手法において、結果の解釈や挙動の制御が困難であるのに対し、

本手法では出力結果は容易に解釈できるものであり、また係り受けの制約を簡単に規定することができ、候補を絞った後の係り先の選択を学習させることができている。今回は少数の単語だけに着目した非常に単純な手法を適用したにすぎないが、LSTM を用いて簡素なモデルを保ちつつ文脈を採り入れる [3] など、今回の手法で考慮できていない文脈を捉えることで改善が見込める。また、明示的に扱っていない並列構造についても、語彙情報やより広い文脈を考慮して解決できるケースもあるので、調査を続けたい。

参考文献

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Danqi Chen and Christopher Manning. 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, 2014.

[3] James Cross and Liang Huang. Incremental parsing with minimal features using bi-directional lstm. In *The 54th Annual Meeting of the Association for Computational Linguistics*, p. 32, 2016.

[4] Timothy Dozat, Peng Qi, and Christopher D Manning. Stanford's graph-based neural dependency parser at the conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pp. 20–30, 2017.

[5] EDR. EDR (Japan Electronic Dictionary Research Institute, Ltd.) electronic dictionary version 1.5 technical guide, 1996.

[6] Hiroshi Kanayama, Kentaro Torisawa, Yutaka Mitsuishi, and Jun'ichi Tsujii. A hybrid Japanese parser with hand-crafted grammar and statistics. In *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 411–417, 2000.

[7] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 523–530. Association for Computational Linguistics, 2005.

[9] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, Vol. 13, No. 2, pp. 95–135, 2007.

[10] Joakim Nivre and Ryan McDonald. Integrating graph-based and transition-based dependency parsers. *Proceedings of ACL-08: HLT*, pp. 950–958, 2008.

[11] Tianze Shi, Felix G Wu, Xilun Chen, and Yao Cheng. Combining global models for parsing universal dependencies. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pp. 31–39, 2017.

[12] Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gökırmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Hěctor Martínez Alonso, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadova, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, 2017.

[13] 金山博. 統計的日本語構文解析器の部分的修正. 情報処理学会第 160 回自然言語処理研究会, pp. 1–8, 2004.

[14] 金山博. 読点に頼らない統計的構文解析. 情報処理学会第 170 回自然言語処理研究会, 2005.

[15] 金山博, 鳥澤健太郎, 光石豊, 辻井潤一. 3 つ以下の候補から係り先を選択する係り受け解析モデル. 自然言語処理, Vol. 7, No. 5, pp. 71–91, 2000.

[16] 白井清昭, 乾健太郎, 徳永健伸, 田中穂積. 統計的構文解析における構文的統計情報と語彙的統計情報の統合について. 自然言語処理, Vol. 5, No. 3, 1998.