逐次型依存構造解析における動的特徴量選択

小比田 涼介　　能地 宏　　松本 裕治
奈良先端科学技術大学院大学 情報科学研究科
{kohita.ryosuke.kj9, noji, matsu}@is.naist.jp

1 はじめに

本稿では、逐次型依存構造解析を対象に、解析器の状態を考慮した動的な特徴量抽出方法について議論する．依存構造の逐次解析は、後続の入力を待たずに漸進的に単語間の依存関係を構築できるため、対話型システムへの応用が期待でき [2]、また、遷移型解析器によって実現可能である [6]．しかしながら、その局所決定的なアルゴリズムはしばしば誤った解析を引き起こす．この問題は、リカレントニューラルネットワークを用いた入力文全体の符号化によって改善されつつあるが [4, 9]、逐次解析との併用はその性質上不可能であるため、別の方法を考える必要がある．

我々が提案する動的な特徴量抽出では、局所情報の最大活用を目的として、フィードフォワードネットにアテンション [1] を組み入れる．図 1 は遷移型解析器が誤りやすい文であるが、先に現れる品詞系列（名詞･助詞･名詞）が典型的な名詞句を想起させ、“*Monday*”の親を“*John*”と誤推定する原因となっている．提案手法が目標とするのは、特徴量抽出時にアテンションを介すことで、当該状態における重要な単語（“*Monday*”）を重み付け、正解の構造（[VP[PP *on Monday*] *introduces*]）を推定できるようなモデルを学習することである．この論文では、標準の遷移型依存構造解析器 [3] を対象に実験を行うが、手法自体は、各種特徴量の連結ベクトルを入力とするような、いずれのフィードフォワードネットにも適用可能である．

実験では、Universal Dependencies（UD）を用いて言語横断的な有効性を検証した結果、多くの言語でアテンションによる精度の向上が確認できた．加えて、人工文を用いた実験によって、アテンションが仮説通りに作用しているかも調査した．具体的には、図 1 のような特定の箇所に着目することで局所曖昧性を解消できる文を解析させ、統制された条件下での性能を検証する．こうした心理言語学的な統制実験 [5] は、近年 NLP のニューラルネットワークモデルの検証においても行われ始めている [8].

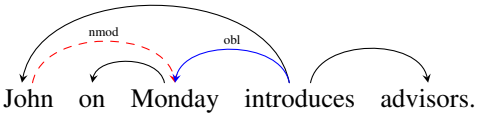

図 1: 局所曖昧性を内包した文．“*Monday*” は “*introduces*” の斜格だが、“*John*”の名詞修飾子として間違えられやすい．

2 モデル

2.1 ベースモデル

提案モデルは Chen&Manning[3] のフィードフォワードネットを用いた遷移型解析器をベースとする[1]．この解析器では、各時刻ごとにスタックとバッファ周辺の単語・品詞・ラベルに関するエンベディングをそれぞれ連結し、ネットワークの入力ベクトルとする（$\mathbf{x}^w$、$\mathbf{x}^p$、$\mathbf{x}^l$）．これらのベクトルを線形写像した後（i.e., $\mathbf{h} = \mathbf{W}^w\mathbf{x}^w + \mathbf{W}^p\mathbf{x}^p + \mathbf{W}^l\mathbf{x}^l + \mathbf{b}$）、非線形変換を経て、最終の softmax 層で各遷移確率を出力する．

2.2 局所特徴量へのアテンション

この章では、入力から隠れ層 $\mathbf{h}$ への特徴量計算において、どのようにアテンションを適用するかについて述べる．$\mathbf{W}^w\mathbf{x}^w$、$\mathbf{W}^p\mathbf{x}^p$、$\mathbf{W}^l\mathbf{x}^l$ の 3 要素は独立しているので、以下 $\mathbf{W}x$ に一般化して説明する．

まず、$\mathbf{x}$ は n 個の特徴量エンベディングを連結したものであるため、$\mathbf{W}$ もまた n 個の要素に分解可能である（図 2）．この時、各要素を $\mathbf{W}_i$ 及び $\mathbf{x}_i$ とした時、$\mathbf{W}\mathbf{x} = \sum_i \mathbf{W}_i\mathbf{x}_i$ が成り立つことに注意されたい．そして、i 番目の特徴量に対応する隠れ層の表現として、$\mathbf{c}_i = \mathbf{W}_i\mathbf{x}_i$ が定義できる．

提案法では、この隠れ層を分解したベクトル $\{\mathbf{c}_i\}$ に対してアテンションを適用する．具体的には、隠れ層をアテンションベクトル $\mathbf{a} = (a_1, a_2, \cdots, a_n)$ による重み付き和として、$\mathbf{h}_g = \sum_i a_i\mathbf{c}_i$ のように計算する．

[1]本実験では、オリジナルに対していくつかの特徴量を追加しているが（3.1 章参照）、簡略化のためここでの説明はオリジナルの特徴量のみを用いて行う．

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

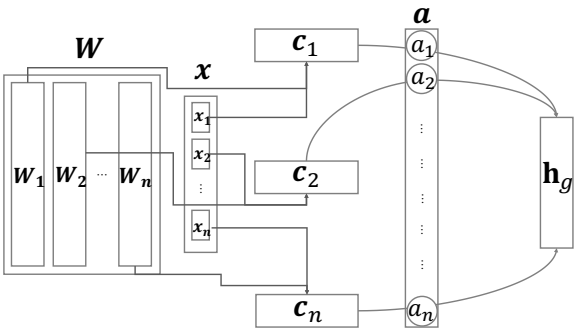

図 2: $\mathbf{W}_i\mathbf{x}_i$ を分解した隠れ層 $\mathbf{c}_i$ に対するアテンション．

アテンション a_i は、$\mathbf{c}_i$ とパラメーター $\mathbf{q}$ を用いて、以下の式で得られる（σ はシグモイド関数）．

$$a_i = \frac{\exp(\sigma(\mathbf{q}\cdot\mathbf{c}_i))}{\sum_{i=1}^{n}\exp(\sigma(\mathbf{q}\cdot\mathbf{c}_i))}$$

アテンションパラメーターは、単語・品詞・ラベルについてそれぞれ異なるもの（$\mathbf{q}^w$、$\mathbf{q}^p$、$\mathbf{q}^l$）を用いる．

3 実験

この章では、UDv2.0[2]における多言語解析の結果（3.2 章）と、人工文を用いた統制実験の結果（3.3 章）を報告する．前述した通り、後者の実験で用いる文は、特定の箇所に着目することで局所曖昧性を解消できるようになっている（3.3.1、3.3.2 章参照）．アテンションが期待通りに作用していれば、それぞれの箇所に重みがかけられ、正しい構造へと辿り着けるだろう．両実験共にアテンションなしの解析器、及び、UDPipe v1.1[3]と比較する．

3.1 解析器

特徴量を抽出してくる解析器上の位置は Chen&Manning[3] に従うが、LEMMA などのオリジナルからの追加情報も使用する．エンベディングのサイズは、WORD を 50、LEMMA、UPOS、XPOS、FEATS、および DEPREL を 20 とした．また、bi-LSTMs を用いた 32 次元の文字エンコーディングも使用した（ただし、アテンションは適用しない）．隠れ層の次元数は 200 で、50%のドロップアウトを適用する．事前学習済みのエンベディングとしては、ベースラインの UDPipe と同じもの [4]を使用し、また、交差構造を扱うために arc-standard swap[7] を採用した．ビームサーチの幅は 5 とした．[5]

	Dev	Test
UDPipe	79.52 (10.13)	70.34 (13.40)
w/o Att.	79.65 (9.38)	70.06 (12.99)
w/ Att.	**80.18** (10.78)	**70.79** (13.19)

表 1: LAS のマクロ平均 (カッコ内標準偏差)

3.2 多言語解析

UDv2.0 の 45 言語 63 ツリーバンクを対象とし、訓練・開発・評価の分割は UDPipe に合わせた．評価は、全ツリーバンクにおける F1LAS 値のマクロ平均とする．開発データではゴールドの文分割・単語分割・形態素情報を用いて解析し、評価データでは生文解析として UDPipe による前処理後の文を解析する．

結果を表 1 に示す．アテンションなしのモデル（w/o Att.）は UDPipe と同等の精度であるが、アテンションの追加（w/ Att.）によって、両方の性能を上回っている．言語ごとに比較すると、開発データでは 54 個、評価データでは 57 個のツリーバンクでアテンションによる精度の向上が確認された．一方、en_partut（1,035 文）や hu（864 文）といった比較的小さいツリーバンクでは性能が低下しており、有効なアテンションの学習にはデータをより多く必要とする可能性がある．

3.3 局所曖昧文の解析

この章では、二つのタイプの人工英文を用いて、局所曖昧性に対してアテンションがどのように役立っているのかを調査する．人工文は、特定の箇所に着目することで曖昧性が解消できるような構造になっており、アテンションが当該箇所に重みをかけて正解に辿り着けるかどうかを検証する．

3.3.1 斜格 vs. 名詞修飾子

最初のセットでは、図 1 のような斜格（oblique, obl）と名詞修飾子（noun modifier, nmod）の曖昧性解消を検証する．このタイプの文では、先行する部分（“*John on Monday*”）の品詞系列が典型的な名詞句と同様であるために、通常の遷移型解析器は “*Monday*” を名詞修飾子と誤認識し、“*John*” にかけてしまうことが多い．

[2]http://hdl.handle.net/11234/1-1983

[3]CoNLL 2017 UD shared task（http://universaldependencies.org/conll17/）において、局所的な特徴量のみを用いた解析器として最高精度を記録している．

[4]https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1990

[5]UDPipe は言語ごとにハイパーパラメーターのチューニングを行っているが、我々の解析器では上記の設定で固定した．ただし、訓練データが非常に少ないカザフ語に関しては、WORD・UPOS・XPOS の次元をそれぞれ 100・50・50 に増やすことで精度が大幅に向上したため、そちらを採用した．

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

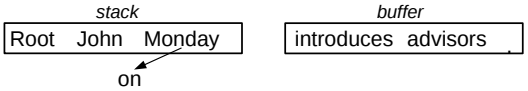

図 3: 遷移型解析器が “*Monday*” の係り関係（斜格 vs. 名詞修飾子）を判別する際の状態

2	3		4		5
	obl	nmod	obl	nmod	
on	**None**	**a**	**Monday**	**balcony**	introduces
ADP	**None**	**DET**	**PROPN**	**NOUN**	VERB
IN	**None**	**DT**	**NNP**	**NN**	VBZ
-	**None**	**Ind ...**	Sing	Sing	Ind ...

表 2: “*John on Monday ...*” (obl) と “*John on a balcony ...*” (nmod) の 2 から 5 番目の単語における差分（形態素情報はゴールド）.

図 3 は “*Monday*” の親を判別する際の遷移型解析器の状態である．解析器の動きとしては、斜格に対応する遷移は shift であるが、ここで誤って right-arc を選択すると前述した不正解の構造が作られる．この時名詞句解釈を誘導するのが “*on*” であるが、「人名詞を修飾することが少なく、動詞と結びつきやすい」といった性質を持つ “*Monday*” に着目することができれば、斜格として後続の動詞句を構築できるだろう．

実験で用いる 14 組の文は、“*John on Monday introduces advisors*”と “*John on a balcony introduces advisors*”のようなミニマルペアになっており、当該の前置詞句は前者では斜格であり（obl)、後者では名詞修飾子となる（nmod)．それぞれの差分を表 2 に示す．

表 3 が各条件における正解数である．典型的な構造を持つ nmod 条件では、いずれのシステムもほぼ完璧に解析できている．一方、obl 条件では、アテンション付きの解析器のみ正答できており、他の解析器はゴールドの形態素情報を与えても失敗している．予測タグを用いた obl 条件において、アテンション付き解析器の精度が低下しているが、これは 5 つ目の動詞（例 “*introduces*”）を名詞とする誤った前処理からのエラー伝搬であった．これらの結果から、一般の遷移型解析器では obl のような文を解析できないことがわかる．そして、アテンションを組み入れることによって、致命的なタグ誤りを含まない限り、こうした局所曖昧性に対処できることも示唆している．

最後に、表 2 の例文について、分岐状態（図 3）における各特徴量への重みを示す（図 4）．obl 条件では、解析器はスタック上の “*Monday*” とバッファ上の “*introduces*” に対して比較的多くの重みをかけている．これは仮説に従う結果であり、アテンションが経験的な直感と一致する形で動作することを示唆している．

	Gold tags		Pred tags	
	obl	nmod	obl	nmod
UDPipe	0 / 14	14 / 14	0 / 14	13 / 14
w/o Att.	0 / 14	14 / 14	0 / 14	13 / 14
w/ Att.	**12** / 14	14 / 14	**6** / 14	13 / 14

表 3: obl と nmod における正解数.

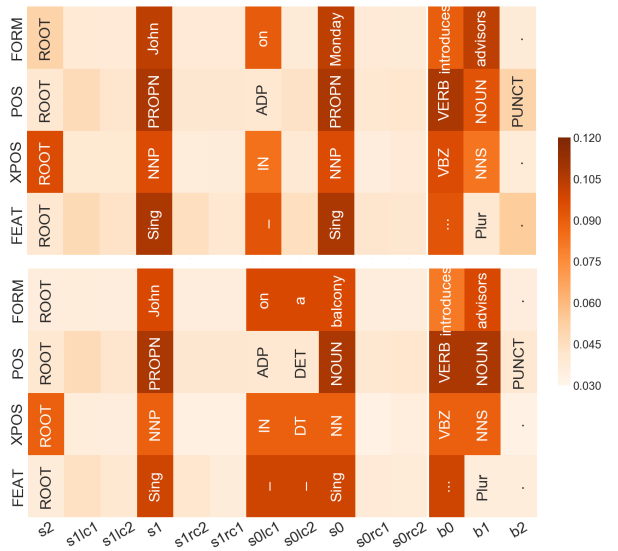

図 4: obl（上図）と nmod（下図）に対するアテンション．s_i 及び b_i はスタックとバッファの i 番目のトークンにそれぞれ対応する．lc_i 及び rc_i はそれらの i 番目の左右の子供に対応する．

3.3.2 目的語—補語 vs. that 節

2 つ目のセットでは、目的語—補語（object-complement, oc）と that 節（that-clause, that）の曖昧性に関する検証を行う．図 5 の上図及び中図がペアとなる例文である．that 条件の文を正しく解析するには、目的語—補語の解釈を避けて、that 省略を察知しなければならない．下図は遷移型解析器の分岐時の状態であり、oc 条件であれば right-arc、that 条件であれば shift が正解遷移である．それぞれの解釈に辿り着くためには、最後のトークンが重要であり、分岐時の状態においてはバッファの 2 番目に位置している．

この実験では 24 組全 48 文を用い、表 4 が各ペア（“*John found it ignored before.*” vs. “*John found it ignored comments.*”）における差分の例である．両者は、品詞タグで明確に区別されるため（例えば、VBN または VBD)、ゴールドのタグが与えられた場合の難易度は高くないと予測される．そのため、この実験における課題は、タグに誤りを含んだ環境であり、特定の箇所や単語表層に着目することで構造を正しく推定できるかどうかである．

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

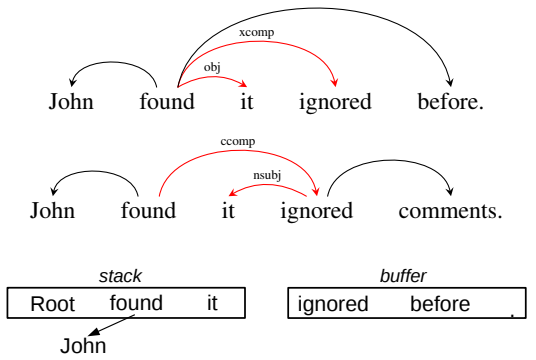

図 5: oc 条件（上図）と that 条件（中図）、及び、遷移型解析器の分岐時の状態（下図）.

3		4		5	
oc	that	oc	that	oc	that
it	it	ignored	ignored	**before**	**comments**
PRON	PRON	VERB	VERB	**ADV**	**NOUN**
PRP	PRP	**VBN**	**VBD**	**RB**	**NNS**
Acc ...	**Nom** ...	**Past** ...	**Ind** ...	_	**Plur**

表 4: 図 5 の文ペアの差分.

	Gold tags		Pred tags	
	oc	that	oc	that
UDPipe	23 / 24	24 / 24	19 / 24	0 (0) / 24
w/o Att.	24 / 24	24 / 24	16 / 24	1 (2) / 24
w/ Att.	23 / 24	24 / 24	18 / 24	1 (7) / 24

表 5: oc と that における正解数．括弧内の数字は、that 省略を察知できたものの他の構造に誤りがあったもの.

結果を表 5 に示す．ゴールドのタグが与えられた場合、いずれのシステムも両者を区別できているが、予測タグにおいては精度が下がり、特に that 条件において顕著である．誤り分析を行うと、これは 3 番目の単語 *it* に対するタグ誤りに起因することが判明した．that 省略によって、UDPipe は “*it*” に対して目的語（Accsuative）のタグを付与しており、さらに、この誤りが次のトークンへのタグ付けにも影響して、4 番目の動詞が過去分詞または形容詞と推定されていた．このようなタグ誤りを内包する文において、解析器が that 省略を察知するのは非常に困難と言える．

しかしながら、完全な解析には失敗しているものの、アテンションを用いた解析器のみが 30 %（7/24）の割合で that 省略を察知できていた[6]．図 6 は、that 省略の察知に成功したケースと失敗したケースのアテンションを可視化したものであり、察知できた場合は最後の名詞に対する重みがわずかに大きくなっている．これが入れ子構造を想定させ、主節の動詞が that 節を取ることを推定できたのだろう．

[6]最後の名詞（例 “*comments*”）を節の軸とし、*it* は名詞の nsubj と解析していた.

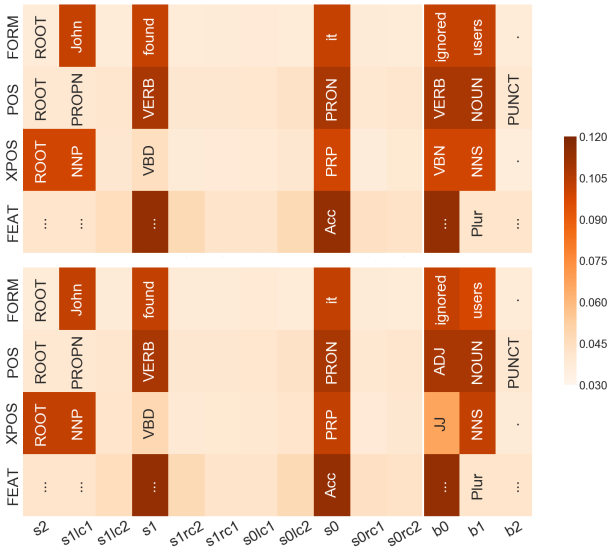

図 6: that に対するアテンション．that 省略の察知に成功した場合（上図）、および、失敗した場合（下図）.

4 結論

本論文では、動的特徴量選択のためのアテンション機構を提案した．提案手法は、特徴量エンベディングを連結したベクトルを入力とするフィードフォワードネットに対して適用可能である．複数の実験を通じて、この仕組みを活用した逐次型解析器は、各々の文脈において着目すべき箇所を学習し、誤りタグに対して頑健になるなど、その性能が向上することを確認した.

参考文献

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. 2015.

[2] Timo Baumann. *Incremental Spoken Dialogue Processing: Architecture and Lower-level Components*. PhD thesis, 2013.

[3] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on EMNLP*, pages 740–750, 2014.

[4] Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327, 2016.

[5] Roger Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):11261177, 2008.

[6] Joakim Nivre. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–554, 2008.

[7] Joakim Nivre, Marco Kuhlmann, and Johan Hall. An improved oracle for dependency parsing with online reordering. In *Proceedings of the 11th IWPT*, pages 73–76, 2009.

[8] Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. Foil it! find one mismatch between image and language caption. In *Proceedings of the 55th Annual Meeting of ACL*, pages 255–265, 2017.

[9] Tianze Shi, Liang Huang, and Lillian Lee. Fast(er) exact decoding and global training for transition-based dependency parsing via a minimal feature set. In *Proceedings of the 2017 Conference on EMNLP*, pages 12–23, 2017.

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.