

複数の項候補を考慮したニューラル文内述語項構造解析

大森 光 小町 守

首都大学東京

omori-hikaru@ed.tmu.ac.jp, komachi@tmu.ac.jp

1 はじめに

述語項構造解析とはテキスト中に存在する述語とその項との意味構造を認識するタスクであり、機械翻訳や含意関係認識など複雑な文脈の解析を必要とする処理のために有用である。その中で文内述語項構造解析は述語と同一文内に存在する項のみを同定するタスクである。例を表1に示す。文内述語項構造解析は、述語と項が直接係り受け関係にある場合 (Dep) と係り受け関係がない場合 (Zero) の解析の2つに分けられる。前者の解析精度は約9割であるのに対し後者は約5割であるため、後者の精度向上に焦点を当てた研究が多くされている [5]。

文内述語項構造解析では、文内に共参照関係や並列構造等特別な文構造が存在しない限り1つの述語に対して2つ以上の同じ必須格 (ガ格, ヲ格, ニ格) を持つことはない。そのため対象の述語のガ格が決定しているときは、文内の項候補からヲ格, ニ格だけを探索すればよい場合がある。

吉川ら [9] は、Markov Logic を利用して、複数の述語項関係の間にある依存関係を考慮する手法を提案した。一方、これまでのニューラルネットワークを用いた文内述語項構造解析モデル [6, 4] では、述語と項候補間の述語項関係を決定するときに、その述語とそれ以外の項候補との述語項関係を明確に考慮していなかった。これらのモデルは述語の同一文内のすべての単語を項候補にしているため、必須格ではない項候補が数多く存在する。我々はある項候補が述語の必須格であるという情報を用いることで、それ以外の大量の項候補が同じ必須格にならないように考慮できると考える。

そこで本研究では、複数の項候補と述語との関係を考慮して、同一述語の必須格を予測するモデルを提案する。提案モデルは、Recurrent Neural Network (RNN) によって、項候補の予測ラベルを次の隠れ層の入力に用いることで、複数の項候補と述語との関係

| | |
|------|--|
| Dep | 今日, 花子は太郎にチョコレートをあげた。 述語: あげる ガ格: 花子 ヲ格: チョコレート ニ格: 太郎 |
| Zero | チョコを貰ったので, 太郎は喜んだ。 述語: 貰う ガ格: 太郎 |

表 1: 文内述語項構造解析の例

を考慮する。

NAIST テキストコーパスを用いた実験の結果、提案モデルは複数の項候補と述語との関係を考慮しないベースラインモデルより高い解析精度を示した。

2 ニューラルネットワークを用いた文内述語項構造解析

近年、文内述語項構造解析においてニューラルネットワークを用いた手法 [3, 7] が優れた解析精度を示している。Ouchi ら [6] は、ある1つの述語項構造情報が他の述語項構造の解析を行う上で役立つと考え、grid-type Recurrent Neural Networks (Grid-RNN) を用いて複数の述語を同時に考慮できるモデルを提案し、特に述語と係り受け関係にない項の解析において解析精度が向上することを示した。Matsubayashi ら [4] は、統語関係埋め込みベクトルなどを使って述語とその項から構成される局所的な素性を表現し、多層 Feed Forward Neural Networks (FFNN) を用いることで、複数の述語とその項の関係を考慮しない局所的なモデルでも高い精度を達成できることを示した。

これらのモデルは述語と同一文内にあるすべての単語を項候補とし、述語とその項候補が述語項関係にあるかを識別する。項候補は文長に比例する個数あるのに対して、述語に対する必須格は高々3つであることから、必須格でない項候補が数多く存在する。これらのモデルではある述語と項候補との述語項関係を決定するときに、その述語と他の項候補との述語項関係は考慮していないが、我々のモデルではある項候補が述

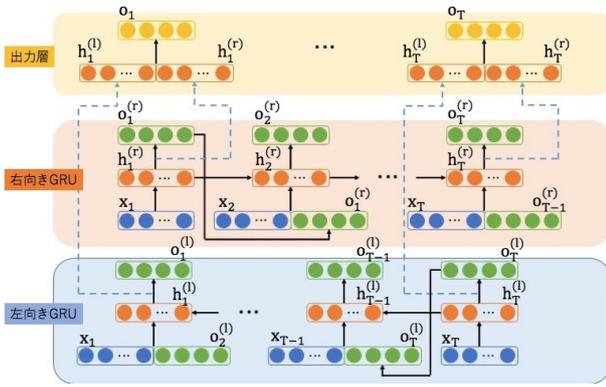


図 1: 提案モデルのネットワーク構造

語の必須格であるという情報を素性として用いることで、それ以外の大量の項候補が同じ必須格になる解析ミスが減らし、また他の必須格を識別する精度を上げることができると考えられる。

3 複数の項候補を考慮したニューラル文内述語項構造解析モデル

提案モデルを図 1 に示す。基本的な構造は Ouchi ら [6] と同様に述語と項のペア素性を RNN に入力し、各隠れ層のベクトルを次元圧縮してソフトマックス関数を用いることで項のラベル $c \in \{\text{ガ格}, \text{ヲ格}, \text{ニ格}, \text{ELSE}\}$ を予測する。

3.1 GRU モデル

文の単語を $\omega_t \in [\omega_1, \omega_2, \dots, \omega_T]$ とする。各単語は単語埋め込み行列によって d 次元の単語ベクトル e_t に変換される。文節ベクトル b_t は ω_t の文節内にある全ての単語ベクトルの平均とし、単語 ω_t を含む文節より 1 つ前の文節ベクトルを b_t^{-1} 、1 つ後の文節ベクトルを b_t^{+1} とする。述語 ω_p の単語ベクトルを e_p とし、述語の文節ベクトルを b_p とする。述語と項の素性ベクトル f_t は以下のように定義される。

$$f_t = e_p \oplus b_p \oplus e_t \oplus b_t^{-1} \oplus b_t \oplus b_t^{+1} \quad (1)$$

\oplus はベクトルの結合を表す。この素性ベクトル f_t とパラメータ行列 $W_1 \in \mathcal{R}^{d \times 6d}$ を掛け合わせ得られたベクトルを RNN の入力ベクトル \mathbf{x}_t とする。

$$\mathbf{x}_t = W_1 f_t \quad (2)$$

RNN は各タイムステップ t において、入力ベクトル \mathbf{x}_t を受け取り、各隠れ層から d_h 次元のベクトル \mathbf{h}_t を出力する。RNN に Gated Recurrent Unit (GRU) [1] を用い、RNN 層は Bi-directional Gated Recurrent Unit (Bi-GRU) で構成する。右向き GRU の隠れ層の

ベクトルを $\mathbf{h}_t^{(r)}$ 、左向き GRU の隠れ層のベクトルを $\mathbf{h}_t^{(l)}$ とする。 $g(\cdot)$ は GRU の処理を表す。

$$\mathbf{h}_t^{(r)} = g(\mathbf{x}_t, \mathbf{h}_{t-1}^{(r)}) \quad (3)$$

$$\mathbf{h}_t^{(l)} = g(\mathbf{x}_t, \mathbf{h}_{t+1}^{(l)}) \quad (4)$$

$$\mathbf{h}_t = \mathbf{h}_t^{(r)} \oplus \mathbf{h}_t^{(l)} \quad (5)$$

その後出力層に各隠れ層のベクトル \mathbf{h}_t を入力し、得られたベクトルをソフトマックス関数を用いて正規化したものを出力ベクトル \mathbf{o}_t とする。

$$\mathbf{o}_t = \text{softmax}(W_2 \mathbf{h}_t) \quad (6)$$

$W_2 \in \mathcal{R}^{4 \times 2d_h}$ はパラメータ行列である。出力ベクトル \mathbf{o}_t の各要素は、項候補が述語のガ格、ヲ格、ニ格であるか、必須格でないかを表す確率値となっており、確率値最大のラベルを予測ラベル y として出力する。モデルパラメータはクロスエントロピー誤差関数に基づいて学習する。

3.2 提案モデル

提案モデルでは、各隠れ層の入力に 1 つ前の隠れ層から計算された出力ベクトルを追加する。

$$\mathbf{h}_t^{(r)} = g(\mathbf{x}_t \oplus \mathbf{o}_{t-1}^{(r)}, \mathbf{h}_{t-1}^{(r)}) \quad (7)$$

$$\mathbf{h}_t^{(l)} = g(\mathbf{x}_t \oplus \mathbf{o}_{t+1}^{(l)}, \mathbf{h}_{t+1}^{(l)}) \quad (8)$$

$$\mathbf{o}_t^{(r)} = \text{softmax}(W_3 \mathbf{h}_t^{(r)}) \quad (9)$$

$$\mathbf{o}_t^{(l)} = \text{softmax}(W_4 \mathbf{h}_t^{(l)}) \quad (10)$$

$W_3, W_4 \in \mathcal{R}^{4 \times d_h}$ はパラメータ行列である。RNN モデルと同様に式 (5) で隠れ層のベクトル \mathbf{h}_t を計算し、式 (6) から最終的な出力ベクトル \mathbf{o}_t を得る。対象の項候補 ω_t のラベルを予測するとき、右向き GRU では項候補 ω_t より左側にある項候補 $\omega_l \in [\omega_1, \dots, \omega_{t-1}]$ の予測結果を考慮し、左向き GRU では項候補 ω_t より右側にある項候補 $\omega_r \in [\omega_{t+1}, \dots, \omega_T]$ の予測結果を考慮している。そのため \mathbf{h}_t は対象の項候補以外の全ての項候補の予測結果を考慮したベクトルとなる。

4 NAIST コーパスを用いた実験

4.1 実験データ

実験データには NAIST テキストコーパス 1.5 [10] を用いた。Taira ら [8] と同様に、NAIST テキストコーパスの 1 月 1 日から 11 日の記事と 1 月から 8 月の社説を学習データに、NAIST テキストコーパスの 1 月 12 日から 13 日の記事と 9 月の社説を開発データに、NAIST テキストコーパスの 1 月 14 日から 17 日の記

事と10月から12月の社説を評価データに使用した。単語境界と文節境界はNAISTテキストコーパスに付与されているCaboCha形式のデータを使用した。

訓練データについては全てのラベルをトレーニングに用いた。開発・評価データについて、同じ必須格の項が文中に2つ以上ある場合の正解データの作り方は、述語と係り受けの関係にある項のみを正解とし、それ以外の項はELSEラベルにラベル付けする。係り受けの関係にある項がない場合は距離 $|\omega_p - \omega_t|$ が最小の項を正解とする。距離が等しい場合は述語より左側の項を正解とする。

4.2 比較手法

Ouchiら[6]のニューラル文内述語項構造解析手法と比較を行う。Ouchiらは述語の周辺 n 単語を文脈素性としてRNNに入力しているが、我々は項候補を含む文節の前後の文節を文脈素性としてRNNに入力している点が入力素性の主な違いである。

Single-Seq: RNN層にstacked Bi-GRUを採用したモデル(Ouchiら[6])

Multi-Seq: RNN層にGrid-GRUを採用し、複数の述語を同時に考慮したモデル(Ouchiら[6])

Single-Arg (ベースライン): 3.1節で説明した複数の項候補を考慮しないベースラインモデル

Multi-Arg: 3.2節で説明した複数の項候補を考慮する提案モデル

提案モデルは深層学習ライブラリchainer(v2.1.0)で実装し、CPU(Intel Xeon E5-2697)上で実行した。学習時間は1エポックあたり約170分であった。

4.3 ハイパーパラメータ設定

単語の埋め込みベクトルの次元数は32次元、GRUの隠れ層の次元数は32次元に設定した。単語埋め込みの重み行列は $[-0.25, 0.25]$ の一様分布に従ってサンプリングした値を初期値として設定した。学習データの中に2回以上出現する単語を単語ベクトルとし、それ以外の単語は未知語ベクトルとして扱う。ミニバッチサイズは8とし、最適化手法にはAdaDelta($\epsilon = 10^{-6}$, $\rho = 0.95$)を適用し、勾配クリッピングは4に設定した。

閾値 θ を設定し、ELSEラベルの予測確率値が θ より小さい場合はELSEラベルの予測確率値を0にする。閾値 θ は開発データで調整する。エポック数は20に設定し、開発データのF値が最も高かったエポックのモデルで評価データを評価する。

4.4 結果

実験結果を表2に示す。提案モデルがベースラインモデルよりも1.25ポイント高い解析精度を記録し、全てのラベルにおいて提案モデルがベースラインモデルよりも高い解析精度を示した。このことから項候補の予測ラベルを次の隠れ層の入力に用いることで、複数の項候補と述語との関係を考慮し解析精度が上昇することが分かった。

しかし先行研究のSimple GRUとベースラインモデルを比較すると3ポイント以上の差があり、係り受け関係の有無・格の違いに関わらず、本研究のベースラインモデルの精度が低かった。

4.5 考察

Ouchiらは入力素性に項候補が述語であるか否かを表す二値素性を用いていたが、我々は単語ベクトルから構成されるベクトルのみを素性として入力している。そのため我々のモデルは明確に述語の位置を捉えることができず、連体修飾を捉えづらいつ考えられる。また、Ouchiらは $\{2, 4, 6, 8\}$ 層のRNN層を持つモデルのうち最も良いものを最終的なモデルとして採用しているが、我々は提案モデルとベースラインモデルを比較するためにRNN層を2層に限定している。提案モデルをより多層のRNN層を持つネットワーク構造に改良することが今後の課題の一つである。

Single-SeqとMulti-Seqを比較すると、ヲ格などの解析精度が低下しているのに対して、Single-ArgとMulti-Argを比較すると、全ての格ラベルで解析精度が上昇している。これは提案モデルがすべての必須格を考慮したことで、事例数の多い格を優先的に出力するようにモデルが学習するのを避けることができたためだと考えられる。

提案モデルでは正しく予測できたがベースラインモデルでは正しく予測できなかった解析例と、ベースラインモデルでも提案モデルでも正しく予測できなかった解析例を表3に示す。成功例では、ベースラインモデルが2つの異なる項をヲ格であると誤って予測しているのに対して、提案モデルは正しいヲ格を予測できている。失敗例では、どちらのモデルも誤ったガ格を予測しているが、提案モデルではガ格の項を2つ出力するのを避けたため、ベースラインモデルと比べて結果的にガ格(Zero)の適合率の低下を防いでいる。これは、提案モデルが同じ必須格を出力するのを制限できたためであると考えられる。

| Method | Dep | | | | | Zero | | | |
|--------------------------|-------|-------|-------|-------|-------|-------|-------|-------|------|
| | ALL | ALL | ガ格 | ヲ格 | ニ格 | ALL | ガ格 | ヲ格 | ニ格 |
| Single-Seq (Ouchi ら [6]) | 81.15 | 88.10 | 88.32 | 93.89 | 65.91 | 46.10 | 49.51 | 35.07 | 9.83 |
| Multi-Seq (Ouchi ら [6]) | 81.42 | 88.17 | 88.75 | 93.68 | 64.38 | 47.12 | 50.65 | 32.35 | 7.52 |
| Single-Arg (ベースライン) | 76.68 | 85.12 | 85.78 | 91.62 | 58.99 | 38.26 | 42.41 | 24.70 | 5.32 |
| Multi-Arg | 77.93 | 85.91 | 86.66 | 92.34 | 60.61 | 41.83 | 45.91 | 29.85 | 6.60 |

表 2: 文内述語項構造解析実験結果 (F 値)

| 成功例 | |
|----------------------|----------------------------------|
| 述語: 焼却する (遺体 ヲ) | 現場は、遺体が切断、焼却された方品村の犬繁殖場から西へ約五キロ。 |
| Single-Arg (ベースライン) | 現場 ヲ:Zero, 村 ヲ:Zero |
| Multi-Arg | 遺体 ヲ:Zero |
| 失敗例 | |
| 述語: 占める (アジア ガ, 割 ヲ) | 貸付残高が最も多い地域はアジアで、全体の八割を占める。 |
| Single-Arg (ベースライン) | 残高 ガ:Zero, 地域 ガ:Zero, 割 ヲ:Dep |
| Multi-Arg | 地域 ガ:Zero, 割 ヲ:Dep |

表 3: 提案モデルの成功例と失敗例

5 おわりに

本研究では、RNN の隠れ層から計算した出力ラベルを次の隠れ層の入力に用いることで、複数の項候補の予測ラベルを考慮した文内述語項構造解析モデルを提案した。提案モデルは RNN を用いたベースラインモデルよりも高い解析精度を達成することを示した。

提案モデルでは項候補の予測ラベルを他の項候補の予測に用いるため、予測ラベルが間違っていた場合は解析ミスを伝播させてしまう問題がある。Easy-First アルゴリズム [2] を用いることで、正しく予測されたラベルから考慮するようなモデルに改善することが今後の課題として挙げられる。

謝辞

奈良先端科学技術大学院大学の内啓樹さんには正解事例の作り方に関して助言をいただきました。ここに感謝の意を示します。

参考文献

- [1] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, pp. 1724–1734, 2014.
- [2] Yoav Goldberg and Michael Elhadad. An efficient algorithm for easy-first non-directional dependency

parsing. In *Proceedings of NAACL*, pp. 742–750, 2010.

- [3] Ryu Iida, Kentaro Torisawa, Jong-Hoon Oh, Canasai Kruengkrai, and Julien Kloetzer. Intra-sentential subject zero anaphora resolution using multi-column convolutional neural network. In *Proceedings of EMNLP*, pp. 1244–1254, 2016.
- [4] Yuichiroh Matsubayashi and Kentaro Inui. Revisiting the design issues of local models for Japanese predicate-argument structure analysis. In *Proceedings of IJCNLP*, pp. 128–133, 2017.
- [5] Hiroki Ouchi, Hiroyuki Shindo, Kevin Duh, and Yuji Matsumoto. Joint case argument identification for Japanese predicate argument structure analysis. In *Proceedings of ACL-IJCNLP*, pp. 961–970, 2015.
- [6] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. Neural modeling of multi-predicate interactions for Japanese predicate argument structure analysis. In *Proceedings of ACL*, pp. 1591–1600, 2017.
- [7] Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. Neural network-based model for Japanese predicate argument structure analysis. In *Proceedings of ACL*, pp. 1235–1244, 2016.
- [8] Hiroto Taira, Sanae Fujita, and Masaaki Nagata. A Japanese predicate argument structure analysis using decision lists. In *Proceedings of EMNLP*, pp. 523–532, 2008.
- [9] 吉川克正, 浅原正幸, 松本裕治. Markov Logic による日本語述語項構造解析. *自然言語処理*, Vol. 20, No. 2, pp. 251–271, 2013.
- [10] 飯田龍, 小町守, 井之上直也, 乾健太郎, 松本裕治. 述語項構造と照応関係のアノテーション: NAIST テキストコーパス構築の経験から. *自然言語処理*, Vol. 17, No. 2, pp. 25–50, 2010.