マルチタスク学習による遷移型依存構文解析

寺西 裕紀　　進藤 裕之　　松本 裕治
奈良先端科学技術大学院大学 情報学研究科
{teranishi.hiroki.sw5, shindo, matsu}@is.naist.jp

1 はじめに

ニューラルネットワークを用いた依存構文解析は，従来の非ニューラルネットワークのモデルによる構文解析の性能を大幅に上回っている．従来の構文解析器では学習に用いる素性として単語・品詞などの組み合わせを人手によって設計していたのに対し，ニューラルネットワークを用いた構文解析のモデルでは単語・品詞の分散表現からモデル自身が有用な特徴を獲得することができる．いずれのモデルにおいても構文解析において品詞は重要な特徴として用いられているが，事前に付与された品詞の誤りの伝搬を解消することができない．本研究では品詞タグ付け・依存構文解析のパイプライン処理に起因する誤りを緩和し，頑健に構文解析を行うことができるマルチタスク学習のモデルを提案する．

自然言語処理において構文解析のみならず意味解析や上位のアプリケーションでは，学習・推論に至るまでに事前に単語分割，品詞タグ付け，構文解析などを順次行い，素性となり得るデータを付与する場合が多い．しかしながら，単語分割や品詞タグ付け，形態素解析といった下位のタスクにおいても，構文・意味的な曖昧性から誤りが生じる．下位のタスクの誤りはパイプライン処理を経る過程で，目的となるタスクに至るまでのタスクでさらに誤りを生じさせる．パイプライン処理による誤り伝搬の影響を低減・回避する方法として，近年では下位のタスクの結果を素性として用いない，ニューラルネットワークによるエンドツーエンドのモデルが提案されてきている．しかし，解析精度の点において構文・意味的な解析の結果を用いる手法に勝るモデルが開発されていないタスクも多い．そこで近年では一つのニューラルネットワークのモデルに複数のタスクを学習させる，マルチタスク学習が注目されている [4, 10, 5]．マルチタスク学習では複数のタスクを同時に学習することでタスク間で共通する抽象的な特徴をとらえ，タスクの精度向上に役立てることができる．またマルチタスク学習を用いる利点としては他に以下が挙げられる [2]．

- 複数のタスクに取り組むことによって学習データが増幅し，各タスクの学習データに観測されるノイズに対して頑健に学習ができる
- 異なる難易度のタスクに対し，容易なタスクから得た特徴によって困難なタスクの学習に活用できる
- あるタスクのみに観測される局所解に陥りにくく，複数のタスク間で共通する局所解が選好される

本研究の貢献は，依存構文解析に有効な品詞タグ付け・構文解析のマルチタスク学習の階層的モデルを開発し，依存構文解析のタスクにおいて品詞タグ付けの精度を損なうことなく，既存の遷移型構文解析器と比較して同等以上の精度を得たことである[1]．

2 提案手法

本節では，本研究に用いるニューラルネットワークのモデルとその学習方法について説明する．**図 1** は本研究で用いるニューラルネットワークのモデルの概要である．モデルは大きく分けて二つの部分から成る．一つは品詞タグ付けのモデル（図中の (a)〜(e) のレイヤー），もう一つは構文解析のモデル（図中の (f)〜(h) のレイヤー）である．次小節以降はモデルの詳細について述べる．

2.1 品詞タグ付けのモデル

モデルの入力として，単語の系列を受け取る．単語および語を構成する文字は語彙数次元の one-hot ベクトルで表され，分散表現のベクトルに変換される（**図 1** (a), (b)）．

[1] 本研究で使用したコードは以下に公開する．
https://github.com/chantera/mtl-parser

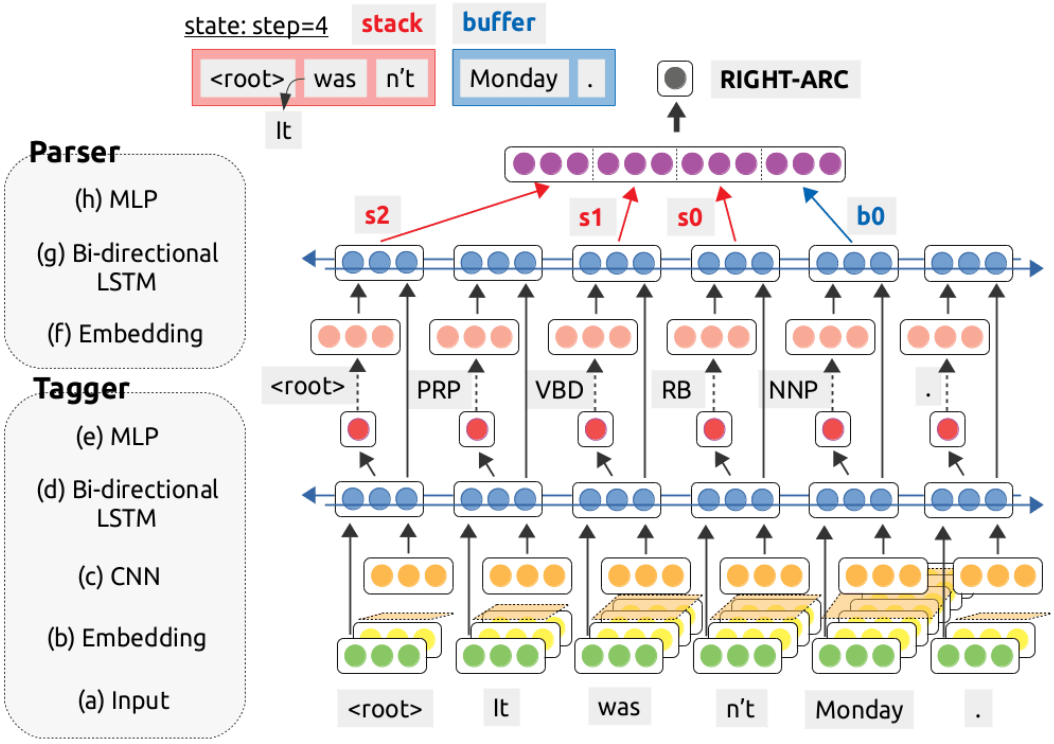

図 1: ニューラルネットワークのモデルの概要

$$\begin{aligned} \mathbf{e}_t^{word} &= W^{word} x_t^{word} \\ \mathbf{e}_{t,i}^{char} &= W^{char} x_{t,i}^{char} \\ E_t^{char} &= \{\mathbf{e}_{t,1}^{char}, \ldots, \mathbf{e}_{t,M}^{char}\} \end{aligned} \tag{1}$$

文字数 M，次元数 d^{char} の文字分散表現のベクトルの系列は，行列 $E_t^{char} \in \mathbb{R}^{d^{char} \times M}$ として CNN の入力として渡され，pooling の演算によってベクトル化される（**図 1** (c)）．

$$\mathbf{r}_t^{char} = pooling(\mathrm{CNN}(E_t^{char})) \tag{2}$$

単語の分散表現のベクトルと文字列の表現ベクトルは連結され，双方向型 LSTM によってベクトル系列の順方向・逆方向から順次計算が行われる（**図 1** (d)）．

$$\begin{aligned} \mathbf{h}_{1:N}^{tag} = \mathrm{BiLSTM}^{tag}([\mathbf{e}_1^{word}; \mathbf{r}_1^{char}], \\ \ldots, [\mathbf{e}_N^{word}; \mathbf{r}_N^{char}]) \end{aligned} \tag{3}$$

品詞タグのスコアは双方向型 LSTM の隠れ状態のベクトルの系列から多層パーセプトロンによって計算され，スコアの最も高い品詞タグをモデルの予測として決定する（**図 1** (e)）．

$$\begin{aligned} \mathbf{s}_t^{tag} &= \mathrm{MLP}^{tag}(\mathbf{h}_t^{tag}) \\ \hat{p}_\theta(y_t^{tag}|x) &= \mathrm{softmax}(\mathbf{s}_t^{tag}) \\ \hat{y}_t^{tag} &= \arg\max_{y_t^{tag}} \left(\hat{p}_\theta(y_t^{tag}|x)\right) \end{aligned} \tag{4}$$

なお，本モデルではそれぞれの品詞タグの決定は独立に行い，CRF などの構造的な学習・推論は用いない．

2.2 依存構文解析のモデル

本研究で用いる遷移型依存構文解析のモデルは，Kiperwasser ら [6] のモデルを基本とする．遷移型の構文解析におけるある状態 state_k に対して，状態 state_k のスタック，バッファから素性に用いるトークンを取り出す．本研究では素性に用いるトークンとして，スタックの上位 3 トークン，バッファの先頭 1 トークンを用いる．それぞれのトークンは解析対象の文における異なる語 word_i を指している．状態 state_k の特徴ベクトルは，トークンが指す語の位置 i を用いて，双方向型 LSTM の出力系列 $\mathbf{h}_{1:N}^{action}$ から隠れ状態ベクトル $\mathbf{h}_i^{action}$ を取り出し，連結することで得る．すなわち，スタックの上位 3 トークン，バッファの先頭 1 トークンが指す語の位置をそれぞれ i_{s_0}，i_{s_1}，i_{s_2}，i_{b_0} とすると，状態 state_k の特徴ベクトル v_k^{action} は次式で表される．

$$\mathbf{v}_k^{action} = [\mathbf{h}_{i_{s_0}}^{action}; \mathbf{h}_{i_{s_1}}^{action}; \mathbf{h}_{i_{s_2}}^{action}; \mathbf{h}_{i_{b_0}}^{action}] \tag{5}$$

状態 state_k の次の状態へと遷移するアクションは，品詞タグ付けと同様に特徴ベクトル v_k^{action} から多層パーセプトロンによってスコア計算を行うことで求める（**図 1** (h)）．

$$\begin{aligned}\mathbf{s}_k^{action} &= \text{MLP}^{action}(\mathbf{v}_k^{action}) \\ \hat{p}_\theta(y_k^{action}|state_k) &= \text{softmax}(\mathbf{s}_k^{action}) \\ \hat{y}_k^{action} &= \arg\max_{y_k^{action}}\left(\hat{p}_\theta(y_k^{action}|state_k)\right)\end{aligned} \tag{6}$$

なお，本モデルではそれぞれのアクションの決定は独立に行い，ビーム探索などの構造的な学習・推論は用いない．また，学習時には gold のアクションによるオラクルの状態に対してアクションの学習を行い，推論時には初期状態から予測のアクションを順次適用させて遷移を繰り返し，終了状態から最終的な構文木を取り出す．

2.3 モデル間の接続

品詞タグ付けのモデルと構文解析のモデルの接続は，品詞タグ付けのモデルにおける双方向型 LSTM を下層とし，構文解析のモデルにおける双方向型 LSTM を上位の層として配置することで実現する．また，品詞タグ付けの結果を用いるために，予測した品詞タグを分散表現のベクトルに変換し，品詞タグ付けのモデルの双方向型 LSTM の出力と連結し，構文解析のモデルの LSTM の入力とする（**図 1** (f), (g)）．

$$\begin{aligned}\mathbf{e}_t^{tag} &= W^{tag}\hat{y}_t^{tag} \\ \mathbf{h}_{1:N}^{action} &= \text{BiLSTM}^{action}(\\ &\qquad [\mathbf{e}_1^{tag};\mathbf{h}_1^{tag}],\ldots,[\mathbf{e}_N^{tag};\mathbf{h}_N^{tag}])\end{aligned} \tag{7}$$

2.4 学習

ニューラルネットワークのパラメータ集合 θ の学習は品詞タグの交差エントロピー誤差と遷移システムのアクションの交差エントロピー誤差の和を確率的勾配法によって最小化することにより行う．

$$\begin{aligned}L(\theta) = &-\sum_{d=1}^{D}\sum_t \log\hat{p}_\theta(y^{tag(t)}|x^{(t)}) \\ &-\sum_{d=1}^{D}\sum_k \log\hat{p}_\theta(y^{action(k)}|state^{(k)})\end{aligned} \tag{8}$$

3 実験

3.1 実験設定

マルチタスク学習の効果を測定するために Penn Treebank version 3 を用いて英語の品詞タグ付けと依存構文解析の実験を行った．コーパスは Stanford Parser version 3.3.0 を用いて Stanford Dependency に変換をし，Wall Street Journal パートのセクション 2 から 21 を訓練データ，22 を開発データ，23 を評価データとして使用した．依存構文解析の遷移システムには ArcStandard，ArcHybrid を用いた．また依存構文解析のアクションの系列は static オラクルに従って抽出した．

単語の分散表現には事前学習した 100 次元のベクトル[2]を用いて初期化を行った．文字の分散表現のベクトルの次元数は 10 次元とし，標準正規分布に従う乱数で初期化をした．CNN の window 幅は 5 とし，出力のチャンネル数は 50 とした[3]．pooling の関数には max-pooling を用いた．双方向型 LSTM は**図 1** (d), (g) のいずれも 2 層にし，隠れ状態のベクトルの次元数は 400 次元とした．品詞の分散表現のベクトルの次元数は 50 次元とし，標準正規分布に従う乱数で初期化した．MLP は**図 1** (e), (h) のいずれも中間層を 1 層にし，ユニット数はそれぞれ 100，800 とし，活性化関数は ReLU を用いた．パラメータの初期化は LSTM のみ直交行列を用いて初期化をし，その他は全て Glorot の初期化を用いた．学習の最適化には Adam を用い，初期の学習率を 0.001 とした．ミニバッチのサイズは 32 とし，勾配の L2 ノルムは閾値を 5 とするクリッピングを適用した．また正則化のために，**図 1** (a)〜(h) のレイヤー間に Dropout（ratio=0.5）を適用した．

全ての実験で提案モデルの訓練を 50 エポック行い，開発データ中で UAS の最も高いモデルを用いて評価データでの評価を行った．また，パイプライン処理による依存構文解析と比較をするために，提案手法の品詞タグ付けのモデルのみを用いて，10 分割ジャックナイフ法で 50 エポックずつ学習を行って品詞タグを付与し，予測の品詞タグを用いて提案モデルにて構文解析のタスクを行った[4]．また，品詞タグ付けの性能を先行研究と比較するために，コーパスの 0 から 18 を訓練データ，19 から 21 を開発データ，22 から 24 を評価データとして用い，提案手法の品詞タグ付けモデルとマルチタスクモデルの実験を行った．

[2] https://nlp.stanford.edu/projects/glove/
[3] padding 処理は Ma ら [8] と同様に PAD トークンを用いた
[4] 付与された品詞タグを使用するため，**図 1** (e) のレイヤーは使用せず，付与したタグを (f) の入力とした．

モデル	手法	UAS	LAS
This work (pipe, AS)	greedy	94.00	91.86
This work (pipe, AH)	greedy	93.93	91.73
This work (mtl, AS)	greedy	94.25	92.45
This work (mtl, AH)	greedy	94.18	92.38
Q & M 2017 [9]	greedy	94.3	92.2
K & G 2016 [6]	greedy	93.9	91.9
C & M 2014 [3]	greedy	91.8	89.6
Andor et al. 2016 [1]	beam	94.61	92.79
Weiss et al. 2015 [11]	beam	94.26	92.41

AS=Arc-Standard，AH=ArcHybrid

表 1: 依存構文解析の実験結果

モデル	accuracy
This work (tagging-only)	97.44
This work (mtl, AS)	97.47
This work (mtl, AH)	97.51
Hashimoto et al. 2017 [5]	97.55
Andor et al. 2016 [1]	97.45
Ma & Hovy 2016 [8]	97.55
Ling et al. 2015 [7]	97.78

表 2: 品詞タグ付けの実験結果

3.2 実験結果

実験結果を**表 1**，**表 2** に示す．依存構文解析の評価実験において，提案手法は既存の遷移型依存構文解析のモデルと比較して高い UAS，LAS のスコアを達成した．Qi ら [9] のモデルは提案手法と同様に品詞タグのマルチタスク学習を行っており，遷移システムについては新たに Arc-Swift というアルゴリズムを提案している．提案手法はパイプライン処理に比べて高い性能を発揮しており，またビーム探索を用いない greedy な遷移型構文解析のなかで最も高い LAS の値を得ている．

品詞タグ付けの評価実験において，品詞タグ付けのみを行ったモデルと比較してマルチタスク学習を行ったモデルでは精度の向上は見られなかった[5]．品詞タグ付けと構文解析の階層的なモデルによるマルチタスク学習では，構文解析を行ったことによる破滅的忘却を生じることなく，品詞タグ付けの精度を維持したまま構文解析の学習を行うことができている．

[5]評価データの品詞タグ付けの精度についてマクネマー検定を行ったところ，マルチタスク-ArcHybrid は有意水準 5%で有意差が得られた．なお，マルチタスクモデルは依存構文解析の評価と同様に開発データ中の UAS が最も高いモデルを選択している．

4 おわりに

本研究では，品詞タグ付けのモデルと依存構文解析のモデルを階層的に接続したモデルを提案し，マルチタスク学習を行った．実験の結果，パイプライン手法と比較して品詞タグ付けの精度を維持したまま，依存構文解析のより高い精度が得られたことを示した．今後は中国語における単語分割のタスクとのマルチタスク学習など，多言語の品詞・形態素・構文解析への応用を行い，またビーム探索などを用いた品詞・構文の構造学習を行う予定である．

参考文献

[1] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[2] Richard Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[3] 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*, 2014.

[4] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2011.

[5] Kazuma Hashimoto, caiming xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

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

[7] Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

[8] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[9] Peng Qi and Christopher D. Manning. Arc-swift: A novel transition system for dependency parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

[10] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[11] David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 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*, 2015.