深層学習による CCG 構文解析と文生成の同時学習

理学専攻・情報科学コース　1840670 馬目華奈　(指導教員：戸次大介)

1　はじめに

自然言語処理の分野において、入力に対し自然言語の文を出力するタスクを文生成と言う。近年の文生成の研究は、ニューラルネットを用いたアーキテクチャである Encoder-Decoder[1] を用いる手法が多くを占めている。具体的には、翻訳や対話など更に細分化されたタスクの目的に応じて Encoder-Decoder を改良し、LSTM 等のニューラルネットをチューニングする文生成の研究が多くみられる。これらの手法から出力結果として得られる文の問題点として、同じ語が繰り返し出力されるなど、明らかに容認性の低い文が出力されてしまうことが挙げられる。

その理由の一つとして、LSTM 等のニューラルネットには、明示的に文法を学習するようには設計されておらず、文法をどの程度学習しているのかは明らかになっていないことがある。一方、このような背景がある中で、文脈自由文法 (Context-Free Grammar: CFG) に基づく構文解析と文生成の同時学習モデルである Recurrent Neural Network Grammars (RNNG) [2] が研究されている。RNNG は、CFG に基づいた Shift-Reduce 法アルゴリズムに基づく構文解析と、それを応用した文生成のアクションを予測するモデルである。

しかしながら、CFG に基づくアクションでは、生成規則数は一般に膨大になる。このため、解析結果である構文木の種類が多くなり、本来ならば入力の文の構文木として正しくないような木も出力されてしまう。これにより RNNG による文生成の際にも文法的に誤りである文も出力される可能性が残ってしまう。そこで本研究では、CFG よりも強い文法的なカバー力を持ち、導出に成功した場合は、文が grammartical であることが保証される組合せ範疇文法 (Combinatory Categorial Grammar: CCG) [3] に基づいた Shift-Reduce 法のアクション予測モデルを提案する。CCG を用いることで、出力される構文木の種類の候補数を絞ることが可能になり、構文木の提案分布がより目標分布に近づくと仮定できる。これにより、さらなる高精度な解析と文生成を目指す。

2　RNNG

2.1　Shift-Reduce 法

RNNG における Shift-Reduce 法のアクションの種類は、非終端記号を導入する NT($\langle category \rangle$)、スタックに記号を移動させる Shift、括弧を閉じる（句の成立を認める）Reduce である。バッファには初期状態では入力文の単語が入っているが、Shift 操作でスタックへと移動し、NT 操作で非終端記号に変換したのち、スタック上の要素に Reduce 操作を適用していくことで、文全体の構文木が構築される。これを応用した文生成では、Shift 操作を、単語予測を行う Gen 操作に代替する。RNNG には、文 x と構文木 y の同時確率 $p(\boldsymbol{x}, \boldsymbol{y})$ を求める generative モデル、文から構文木の確率 $p(\boldsymbol{y}|\boldsymbol{x})$ を求める discriminative モデル、それらから重点サンプリングを用いて求められる言語モデル $p(\boldsymbol{x})$ がある。

2.2　generative モデル

generative モデルは、同時確率 $p(\boldsymbol{x}, \boldsymbol{y})$ を求めるモデルである。$p(\boldsymbol{x}, \boldsymbol{y})$ は、実行されるアクションを (a_i) とする。各タイムステップ t に至るまでの計算を埋め込んだベクトル $(\boldsymbol{u}_t)$ によってパラメータ化された遷移モデルであり、下記のように定義されている。

$$
\begin{aligned}
p(\boldsymbol{x}, \boldsymbol{y}) &= \prod_{t=1}^{|a(x,y)|} p(a_t|a_1, ..., a_{t-1}) \\
&= \prod_{t=1}^{|a(x,y)|} \frac{exp\, \boldsymbol{r}_a^\top \boldsymbol{u}_t + b_{a_t}}{\sum_{a' \in A_G(T_t, S_t, n_t)} exp\, \boldsymbol{r}_{a'}^\top \boldsymbol{u}_t + b_{a'}}
\end{aligned}
$$

アクション系列の埋め込みベクトルを $\boldsymbol{r}$、バイアスベクトルを b とする。時刻 t での状態 $\boldsymbol{u}_t$ は出力バッファ、スタック、アクション履歴の 3 つのデータ構造の埋め込みベクトルを組み合わせて計算する。ベクトル $\boldsymbol{o}_t$ を出力バッファ (T_t)、ベクトル $\boldsymbol{s}_t$ をスタック (S_t)、ベクトル $\boldsymbol{h}_t$ をモデルが $t-1$ までに実行したアクションの履歴 $(a_{<t})$ をそれぞれ埋め込んだものとして、$\boldsymbol{u}$ は、

$$
\boldsymbol{u}_t = tanh\left(\boldsymbol{W}[\boldsymbol{o}_t; \boldsymbol{s}_t; \boldsymbol{h}_t] + c\right)
$$

と定義する。ここで W と c はパラメータである。

2.3　discriminative モデル

discriminative モデルは、入力文を $\boldsymbol{x}$、出力の構文木を $\boldsymbol{y}$ とした時、$p(\boldsymbol{y}|\boldsymbol{x})$ を求めるモデルである。generative モデルと異なる点として、出力バッファT を用いる代わりに、入力文の単語が格納されるバッファB を用いる。

2.4　重点サンプリング

言語モデル $p(\boldsymbol{x})$ を求めるため、重点サンプリングを行う。重み $w(\boldsymbol{x}, \boldsymbol{y})$ は、目標分布 $p(\boldsymbol{x}, \boldsymbol{y})$ と提案分布 $q(\boldsymbol{y}|\boldsymbol{x})$ の比 $p(\boldsymbol{x}, \boldsymbol{y})/q(\boldsymbol{y}|\boldsymbol{x})$ として定義する。

$$
\begin{aligned}
p(\boldsymbol{x}) &= \sum_{\boldsymbol{y} \in Y(x)} p(\boldsymbol{x}, \boldsymbol{y}) \\
&= \sum_{\boldsymbol{y} \in Y(x)} q(\boldsymbol{y}|\boldsymbol{x}) w(\boldsymbol{x}, \boldsymbol{y}) \\
&= \mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x})} w(\boldsymbol{x}, \boldsymbol{y})
\end{aligned} \tag{1}
$$

$p(\boldsymbol{x}, \boldsymbol{y})$ を求めるため、提案分布を用いてエントロピーの計算を行い、構文木を得るため、q の分布から N 回のサンプリングを行う。

$$
\begin{aligned}
\boldsymbol{y}^{(i)} &\sim q(\boldsymbol{y}|\boldsymbol{x}) \;\; i \in \{1, 2, ..., N\} \\
\mathbb{E}_{q(\boldsymbol{y}|\boldsymbol{x})} w(\boldsymbol{x}, \boldsymbol{y}) &\overset{MC}{\approx} \frac{1}{N} \sum_{i=1}^{N} w(\boldsymbol{x}, \boldsymbol{y}^{(i)})
\end{aligned}
$$

$\boldsymbol{y}$ の推定値 $\hat{y}$ は、同時確率 $p(\boldsymbol{x}, \boldsymbol{y})$ のもとで最も高い確率で生成された構文木からサンプリングによって求める。

3 提案手法

CCG に基づく構文木と文生成の同時学習モデルを提案する。

3.1 CCG

CCG は語彙化文法の一種で、近年ではニューラルネットを用いた CCG 構文解析の技術が発展し、高精度な解析を行うことができる [4]。各語には統語範疇が割り当てられ、語と語の統語的・意味的な関係を関数適用や関数合成などの組合せ規則により計算する。統語範疇は CFG のラベルに比べ、豊富な情報を持ち、その種類も多い。CCG は、等位接続構文など複雑な文の統語解析において利点があることが知られている。

3.2 CCG 構文解析と文生成の同時学習モデル

CCG 木の導出を人手でアノテーションしたデータセットである CCGBank[5] の正解の木から、組み合わせ規則を適用可能なカテゴリのルールテーブルを生成する。ルールテーブルを参照し、Reduce 操作と NT($\langle category \rangle$) 操作を許可するか否かに関して制限を加える。

3.3 カテゴリ数の制限

CCGBank のカテゴリ数は 1639 個と膨大であるため、NT($\langle category \rangle$) 操作においてカテゴリを特定することが困難である。そこで、CCG 構文解析器 depccg[4] において用いられているカテゴリ 526 個に限定して、NT($\langle category \rangle$) 操作のアクション予測を行う。

4 実験

4.1 実験設定

CCGBank[5] の WSJ§2-21・§24・§23 をそれぞれ教師（39604 文）・開発（1338 文）・評価（2407 文）データとする。教師データにより与えられたアクション数は 528 個、終端記号数（語彙数）は 54012 個、非終端記号数は（528 個から Shift・Reduce アクションを除いた）526 個である。

4.2 評価方法

WSJ コーパス §23 を用いて評価を行う。構文解析の評価には、RNNG の評価においても用いられているラベルなし F 値を用いる。言語モデルの評価にはパープレキシティ (perplexity: PPL) を用いる。評価データセットを $D = \{\boldsymbol{X}^{(n)}\}$、$n$ 番目の系列の長さを $T^{(n)}$ とする。本研究では底は e を用いる。

$$
\begin{aligned}
PPL &= e^{z} \\
z &= -\frac{1}{N}\sum_{n=1}^{|D|}\sum_{t=1}^{T^{(n)}+1} \log_e P_{model}(\boldsymbol{x}_t^{(n)}, \boldsymbol{X}_{[1,t-1]}^{(n)})
\end{aligned}
$$

4.3 実験結果・考察

解析の結果を表 1 に示す。(D) は discriminative モデル、(G) は generativre モデルを用いた結果である。提案手法の上界は depccg によって作成したアクション系列である。CCG に基づいたアクション予測モデルに加え、カテゴリを高頻度に出現するものに限定したが、disrciminative モデル、generative モデルともに RNNG モデルを下回る結果となった。ただし、Shift-Reduce 法で解析を行う際にに beam 探索を用いていない結果である。また、提案手法のラベルなし評価における再現率・適合率を表 2 に示す。

表 1: WSJ §23 における評価

手法	F 値
RNNG(D)	89.8
RNNG(G)	92.4
depccg	94.0
提案手法 (D)	87.4
提案手法 (G)	88.8

表 2: 構文解析評価

手法	適合率	再現率	UF
提案手法 (D)	87.40	87.43	87.41
提案手法 (G)	88.86	88.79	88.82

言語モデルの結果を表 3 に示す。現状では RNNG、LSTM より PPL の値が高くなっているが、精度向上は今後の課題とする。

表 3: 言語モデルの評価

手法	PPL
5-gram	169.3
LSTM	113.4
RNNG	102.4
提案手法	154.6

5 おわりに

本研究では、文法を学習するニューラルネットである RNNG を用いて、CCG に基づく構文解析と文生成の同時学習モデルを提案した。CCG 構文解析器である depccg によってカテゴリを制限し、CCG の導出ルールに制限を加え学習を行った。現時点での精度は解析・生成ともに十分といえるものではないが、今後は beam 探索を用いることで構文解析・生成の精度を上げていきたい。

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 the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014.

[2] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. In *Proc. of ACL*, pp. 199–209, 2016.

[3] Mark Steedman. Surface Structure and Interpretation. In *The MIT Press*, 1996.

[4] Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. A* CCG Parsing with a Supertag and Dependency Factored Model. In *Proc. of ACL*, pp. 277–287, 2017.

[5] Julia Hockenmaier and Mark Steedman. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn treebank. *Computational Linguistics*, Vol. 33, No. 3, pp. 355–396, 2007.