木構造自己注意機構を用いた教師なし統語構造解析

成田百花 [1]　持橋大地 [2]　小林一郎 [1]
[1] お茶の水女子大学大学院　[2] 統計数理研究所
{g1820529,koba}@is.ocha.ac.jp　daichi@ism.ac.jp

概要

X (旧 Twitter) などのソーシャルメディアでは，口語に近い文体で文章が記述される傾向にある．こうした文は文法規則にとらわれない自由な文体で，構文構造について従来のルールベースの手法では解析することができない文が数多く存在する．そこで本研究では，実データの例として X コーパスを対象に，深層学習を用いて統語構造解析を行う教師なし学習モデルを提案する．提案手法では，条件付き確率場（CRF）による構文木の解析精度の向上を促すよう，Transformer の自己注意機構を入力文の統語構造を反映するように変更したエンコーダを用いる．提案モデルを用いることで，従来のモデルと比較して X コーパスにおける構文解析精度が向上することが確認された．

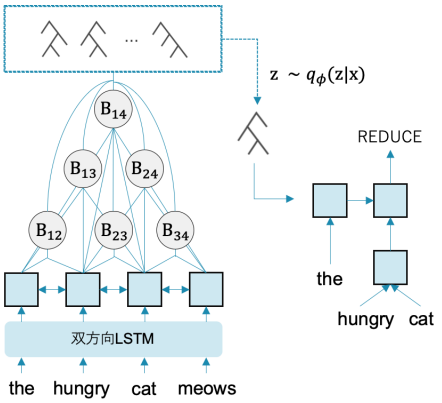
1　はじめに

様々な文字媒体が存在する今日において，それに伴い様々な文体を持つ文が存在している．例えば，X（旧 Twitter）における文は，これまで統語構造解析の主な対象となっていた英字新聞コーパスなどと比べて，文法規則にとらわれない自由な文体を持つことが挙げられる．文法規則にとらわれないコーパスでは，従来の構文規則によるルールベースの手法を用いるだけでは，構文構造を解析することができない文が数多く存在することとなる．一方で，Transformer [1] は高い汎用性を持つことから機械翻訳にとどまらず，様々な自然言語処理課題に対して採用され，高い成果を挙げている．自己注意機構とは，文中の各単語間の関連性を表すスコアを算出するもので，潜在的に統語構造情報を捉えている可能性が考えられることから，これを工夫した統語構造解析手法が提案されている [2] [3]．深層学習を統語構造解析に用いることで， 文を連続値の情報であるベクトルで扱うことが可能となり，従来の文法規則のマッチングによる処理とは違い，より頑健な処理を実現できるという利点がある．そこで本研究では，X コーパスをはじめとするソーシャルメディアのような実データに対応するような自己注意機構を用いた教師なし統語構造解析モデルを提案する．

2　関連研究

Recurrent Neural Network Grammar（RNNG）[4] は, 文とその句構造をトップダウン，また左から右に同時に生成する生成モデルの一つである．Unsupervised Recurrent Neural Network Grammars（URNNG） [5] は RNNG の，アノテーションされた構文木が必要であるという問題点に着目して開発された教師なし学習モデルである．URNNG は，CRF 構文解析器を用いた推論ネットワークと，Stack LSTM [6] と Tree LSTM [7] [8] を用いた生成モデルを組み合わせたアーキテクチャを持ち，変分推論 [9] を用いて学習される．推論ネットワークの CRF には，双方向 LSTM を用いて抽出された特徴量が入力される．しかし，双方向 LSTM の出力には統語構造情報は明示的には反映されていないため，CRF 構文解析器への入力としてはまだ改善の余地がある．そこで本研究では，より確かな統語構造情報を反映するような制約を用いて抽出した特徴量を CRF 構文解析器へ与える手法を提案する．

自己注意機構を用いた教師なし統語構造解析モデルとして，Wang ら [3] は Tree Transformer を提案している．Tree Transformer は Transformer のエンコーダを用いたモデルで，自己注意機構に入力文の統語構造情報を反映するように制約を加える．エンコーダは，BERT [10] の学習タスクである，[Mask] トークンに置き換えられた文中の単語を予測する Masked Language Modeling（MLM) により教師なし学習される．しかし，この MLM による学習は，エンコーダのみを学習の対象としているため，デコーダから出力される構文木について予測をするのではなく，エンコーダからの出力を用いた [Mask] トークンの予測をしており，統語構造の学習が十分とは言えな



推論ネットワーク $q_\phi(\mathrm{z}|\mathrm{x})$　　生成モデル $p_\theta(\mathrm{x},\mathrm{z})$

図 1: URNNG 概要図．（左）CRF 構文解析器を用いた推論ネットワーク（右）RNNG を用いた生成モデル

い．よって，本研究はエンコーダのみの学習ではなく，エンコーダとデコーダ両方をアーキテクチャに持った教師なし統語構造解析モデルを提案する．

3　基本モデル

本節では，提案モデルのベースとなる URNNG [5] のアーキテクチャについて説明する．各単語を x，長さ T の文を $\mathrm{x} = [x_1, ..., x_T]$，長さ T の文に対するラベルなし二分木を $\mathrm{z} \in \mathcal{T}_T$ として，概要図を図 1 に示した．図の左側の推論ネットワークでは，双方向 LSTM によって抽出された特徴量を用いて，CRF 構文解析器で図の点線に示す二分木の分布を生成する．また，図の右側の生成モデルは RNNG [4] であり，Stack LSTM [6] と Tree LSTM [7] [8] で構成され，入力文を復元するように文の生成を行う．

3.1　変分推論

URNNG は変分推論 [9] によって，事後分布 $q_\phi(\mathrm{z}|\mathrm{x})$ を学習可能なパラメータ ϕ を用いて定義する．学習は以下の式 (1) を最適化することを目的とする．

$$\max_{\theta,\phi} \quad \log p_\theta(\mathrm{x}) - \mathrm{KL}[q_\phi(\mathrm{z}|\mathrm{x})||p_\phi(\mathrm{z}|\mathrm{x})] \tag{1}$$

はじめに，事後分布 $q_\phi(\mathrm{z}|\mathrm{x})$ を生成する CRF 構文解析器への入力を，入力文の単語埋め込みと位置埋め込み加えたベクトルに対する双方向 LSTM の出力から得る．双方向 LSTM の出力より前方表現 $[\overrightarrow{h}_1, ..., \overrightarrow{h}_T]$ と後方表現 $[\overleftarrow{h}_1, ..., \overleftarrow{h}_T]$ を隠れ表現として，x_i から x_j のスパンスコア $s_{i,j}$ は式 (2) で表される．

$$s_{i,j} = \mathrm{MLP}([\overrightarrow{h}_1, ..., \overrightarrow{h}_T; \overleftarrow{h}_1, ..., \overleftarrow{h}_T] \tag{2}$$

図 1 内の B は構文木のバイナリ行列であり，x_i と x_j が同じ構成要素に属するときに $\mathrm{B}_{i,j} = 1$，属さないときに $\mathrm{B}_{i,j} = 0$ を表す．そして，CRF 構文解析器がギブス分布によりバイナリ構文木の分布を分割関数 (3) を用いて，式 (4) のように定義する．

$$Z_T(\mathrm{x}) = \sum_{\mathrm{B}' \in \mathrm{B}_T} \exp(\sum_{i \le j} \mathrm{B}'_{ij} s_{ij}) \tag{3}$$

$$q_\phi(\mathrm{B}|\mathrm{x}) = \frac{1}{Z_T(\mathrm{x})} \exp(\sum_{i \le j} \mathrm{B}_{ij} s_{ij}) \tag{4}$$

最後に，双射影 $f : \mathrm{B}_T \to \mathcal{T}_T$ より，$q_\phi(\mathrm{z}|\mathrm{x}) \triangleq q_\phi(f^{-1}(\mathrm{z})|\mathrm{x})$ から事後分布が定義される．

3.2　文の生成

URNNG の生成モデルでは，パラメータ θ を用いて同時確率 $p_\theta(\mathrm{x},\mathrm{z})$ を定義する．モデルは，スタック表現を利用して SHIFT または REDUCE のアクションをサンプリングする．スタックの各要素は，Stack LSTM の隠れ状態と入力ベクトルのペアで構成され，初期スタックを $S = [(0,0)]$，一番上の要素を $\mathrm{top}(s) = (h_{prev}, g_{prev})$ として表す．各ステップのアクション z_t（SHIFT または REDUCE）は，式 (5) よりベルヌーイ分布からサンプリングされる．

$$z_t \sim \mathrm{Bernoulli}(p_t), \quad p_t = \sigma(\mathrm{w}^\top h_{prev} + b) \tag{5}$$

1. $z_t = 0$（SHIFT）の場合
   はじめに，式 (6) で表されるソフトマックス関数を用いたカテゴリ分布より終端記号（単語）を生成する．

$$x \sim \mathrm{softmax}(\mathrm{W}h_{\mathrm{prev}} + \mathrm{b}) \tag{6}$$

   そして，生成された単語 x は Stack LSTM を用いてスタック上にシフトされる．
2. $z_t = 1$（REDUCE）の場合
   スタックから最後の要素を 2 つ取り出し，Tree LSTM を用いて組み合わせ，新しい表現を得る．

生成モデルのパラメータ θ は，Stack LSTM と Tree LSTM のパラメータである．

4　提案モデル

URNNG をベースに，双方向 LSTM の代わりに Tree-Transformer [3] の木構造自己注意機構（4.1 節）を用いたモデルを提案する．また，木構造自己注意

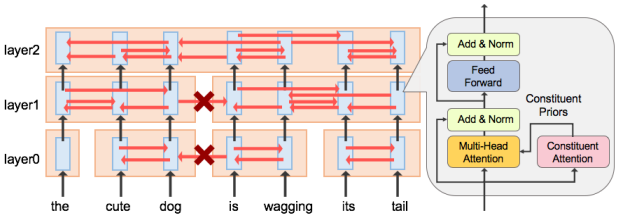


図 2: （左）木構造自己注意機構を用いたエンコーダ．ブロックは入力文から生成された constituent を表している．層を登るにつれ constituent はマージされ徐々に大きくなる．（右）エンコーダ各層の構成．

機構を用いるにあたり，スパンスコアの算出方法についても変更を加える（4.2 節）．

4.1 木構造自己注意機構

URNNG のエンコーダ部がより確かな統語構造情報を抽出するために，木構造自己注意機構を導入する．木構造自己注意機構は，自己注意機構に隣り合う単語間の関連性を捉える “Constituent Attention モジュール”を導入している．“Constituent Attention モジュール”では，N 個の単語を含む文の各単語に対して，文中の句または節（constituent）の区切りを推測する確率 $a = \{a_1, ...a_N\}$ を生成する．確率 a を用いて，同じ constituent に属する単語同士のみで自己注意機構を作用させる制約 “Constituent Priors $C_{i,j}$”が生成される．“Constituent Priors $C_{i,j}$”は，式 (7) で表される，文中の i 番目の単語と j 番目の単語が同じ constituent に含まれるかの確率である．

$$C_{i,j} = \prod_{k=i}^{j-1} a_k \tag{7}$$

上層に移るにつれ constituent は隣同士でマージされて徐々に大きくなり，エンコーダ内部で入力文の構文木を形作る（図 2）．式 (7) を用いて，木構造自己注意機構は式 (8) のように求められる．

$$\text{Attention}(Q, K, V)_{tree} = C \odot \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{8}$$

木構造自己注意機構を用いることで，通常の自己注意機構よりも統語構造情報を反映した特徴量を抽出することができる．これにより，CRF 構文解析器が効率よく文中に含まれる句または節を捉えることを可能とする．

4.2 スパンスコア

URNNG では，CRF 構文解析器に入力するスパンスコアを双方向 LSTM の出力より前方表現 $[\overrightarrow{h}_1, ..., \overrightarrow{h}_T]$ と後方表現 $[\overleftarrow{h}_1, ..., \overleftarrow{h}_T]$ を隠れ表現として用いていた．しかし，提案モデルでは双方向 LSTM とは異なり，木構造自己注意機構により構文情報を含んだベクトルを既に得ていることから，前方表現と後方表現を使った隠れ表現は不要であると考えられる．そこで提案モデルでは，式 (9) のように木構造自己注意機構から出力されたベクトルをそのまま用いることによってスパンスコア s_{ij} を表す．

$$s_{i,j} = \text{MLP}([h_{tree}; h_{tree}]) \tag{9}$$

h_{tree} は木構造注意機構の出力を指す．

表 1: 事前処理後の X コーパスの例文．

where are @person and @person
nooo i dont want to get out of bed
what is ur great m0ment in cricket
in the bus with my twins @url there ’s me

5 実験

本節では，X コーパスの作成手順を示し，4 章で提案したモデルの性能および有効性を X コーパスを用いた実験により検証する．双方向 LSTM を用いた通常の URNNG と，また木構造自己注意機構を用いたスパンスコアを変更する前のモデル（URNNG+Tree）と，提案モデル（URNNG+Tree+Span）で構文解析結果の比較を行った．

5.1 X コーパス

英語で記述された X コーパスを用いる．事前処理を行う前の実データを図 4 に示す．実データを以下の手順で事前処理を行い，コーパスを作成する．

- 複数行に改行された文を削除
- 本文前後のシンボルを削除
- 文の先頭にあるメンションを削除
- メンション、URL を@person, @URL に置換
- 絵文字を含む文を削除
- ハッシュタグを#hash に置換
- 1 単語もしくは 2 単語で構成される文を削除
- 同じ文字の連続を 3 文字に制約
- 全て小文字にする
- 文章を文に分割

事前処理を行った後のデータに含まれる例文を表 1 に示す．さらに学習効率を上げるために事前処理を行った実データに対して，文に含まれる単語数の制限を加える．コーパスに用いられる文は 3〜15 単語で構成されるものに限定し，それぞれ訓練デー

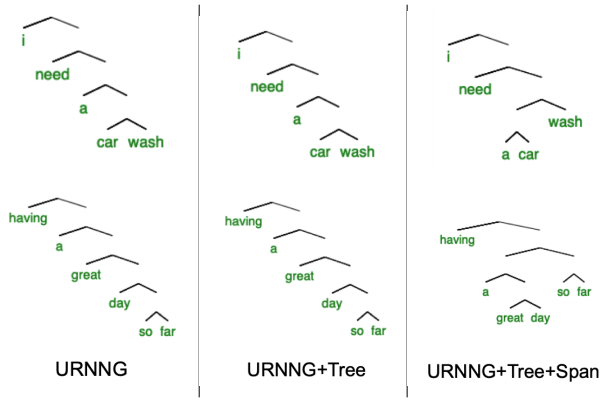


図 3: 双方向 LSTM を用いた通常の URNNG，木構造自己注意機構を用いたスパンスコアを変更する前のモデル（URNNG+Tree），提案モデル（URNNG+Tree+Span）におけるそれぞれの構文解析結果例.

タ 26,987，評価データ 4,475 であり，単語数ごとのデータ数を表 10 に示す.

5.2 実験設定

X コーパスは教師データを持たないため，実験では予測される構文木が正しく解析されているか目視によって評価を行う．入力文の単語埋め込みを作成する際に使用する Vocabulary には，コーパスに 2 回以上現れる単語のみを登録し，Vocabulary 数は 7,330 となった．学習の際，Epoch を 15，batch 数を 16 とする．また，提案モデルにおいて木構造自己注意機構を 10 層，Multi-Head 数を 8 に設定した.

5.3 実験結果

双方向LSTMを用いた通常のURNNG, 木構造自己注意機構を用いたスパンスコアを変更する前のモデル（URNNG+Tree), 提案モデル（URNNG+Tree+Span）それぞれの実験結果を示し，考察を行う．解析結果の例を図 3 と付録に示す．解析結果の描画には Syntax Tree Generator[1)] を用いた.

5.3.1 URNNG

3〜7 単語の文に対して，ほとんどの文が全ての単語の間で括弧が挿入されるような解析（図 3 参照）が行われていた．これは、双方向 LSTM が少ない単語の文の依存関係を正しく抽出することができておらず，モデルのパラメータがうまく学習できていないと考えられる．8 単語以上の文に対しても，全ての単語の間で括弧が挿入されるような解析が多くの文において散見され，課題の残る結果となった.

5.3.2 URNNG+Tree

URNNG と比べて数は減ったが，まだ 3〜7 単語の文おいて全ての単語の間で括弧が挿入されるような解析が行われる文が多く見られる結果となった．しかし，構文木全体として正しい解析は行われていないものの，URNNG よりも部分的に統語構造を捉えようとするような解析が見られた．また，10 単語以上の長い文に対しては正しい解析が散見され，URNNG と比べて解析精度が向上したことが確認できたが，全ての単語の間で括弧が挿入されるような解析が行われる結果もまだ多く見られた.

5.3.3 URNNG+Tree+Span

3〜7 単語の文おいて全ての単語の間で括弧が挿入されるような解析が行われる文の数が URNNG, URNNG+Tree と比べてかなり減少し，多くの文で正しく構文木を解析することができていた．10 単語以上の長い文に対しても，全ての単語の間で括弧が挿入されるような解析が行われる文の数は減少し，部分的に統語構造を正しく捉えているような解析がかなり増加した．これによって，木構造自己注意機構は CRF 構文解析器のパフォーマンスを向上させ，さらに，スパンスコアを変更するこで変更前では重複していた統語構造情報が正しく CRF 構文解析器へ入力されたことが考えられる.

6 まとめ

本論文では，X コーパスを解析する自己注意機構を用いた教師なし統語構造解析モデルを提案した．木構造自己注意機構を用いることで，CRF 構文解析器のパフォーマンスを向上させ，従来の双方向 LSTM を用いたモデルを上回る結果を実験により確認した．加えて，スパンスコアの算出方法を変更することにより，より正しい統語構造情報を CRF 構文解析器へ入力することができることも実験により示された．今後の展望として，入力文の単語埋め込み表現を BERT を用いて得たり，既存の構文解析データ Penn Tree Bank を用いた教師あり学習と組み合わせた半教師あり学習など，モデルの更なる性能向上に取り組みたい.

1） http://mshang.ca/syntree/

謝辞

本研究は JSPS 科研費 18H05521 の助成を受けたものです.

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Proc. of the 31st NIPS**, 2017.

[2] Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2676–2686, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[3] Yau-Shian Wang, Hung yi Lee, and Yun-Nung (Vivian) Chen. Tree transformer: Integrating tree structures into self-attention. In **EMNLP**, 2019.

[4] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. Recurrent neural network grammars. **CoRR**, Vol. abs/1602.07776, , 2016.

[5] Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. Unsupervised recurrent neural network grammars. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 1105–1117, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[6] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. Transition-based dependency parsing with stack long short-term memory. In Chengqing Zong and Michael Strube, editors, **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 334–343, Beijing, China, July 2015. Association for Computational Linguistics.

[7] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In Chengqing Zong and Michael Strube, editors, **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 1556–1566, Beijing, China, July 2015. Association for Computational Linguistics.

[8] Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. Long short-term memory over tree structures. **CoRR**, Vol. abs/1503.04881, , 2015.

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

A　事前処理前の実データ

```
<tweet from="asvparke">
Noooo I dont want to get out of bed !!! :(
</tweet>
<tweet from="genohsis">
@realistbalti princess probs 💁
</tweet>
<tweet from="NikolaWirkus">
@MariieBaumann lmao this was done by you? http://t.co/gl0JROSZl8
</tweet>
<tweet from="_KayxJay">
@YouCraveTati_ thank you Tati ‼️❤️
</tweet>
<tweet from="DatprettyMafaca">
Good Morning @_Candy_N_Vodka ☁️😊
```

図 4: 事前処理前の実データ.

B　事前処理後の X コーパス

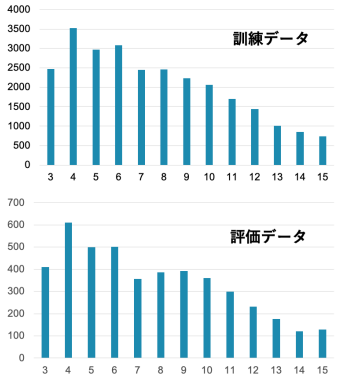


図 5: X コーパスに含まれる文の単語数による内訳.（上）訓練データ.（下）評価データ.

C　構文解析結果例

左から双方向 LSTM を用いた通常の URNNG，木構造自己注意機構を用いたスパンスコアを変更する前のモデル（URNNG+Tree），提案モデル（URNNG+Tree+Span）におけるそれぞれの構文解析結果例を示す.

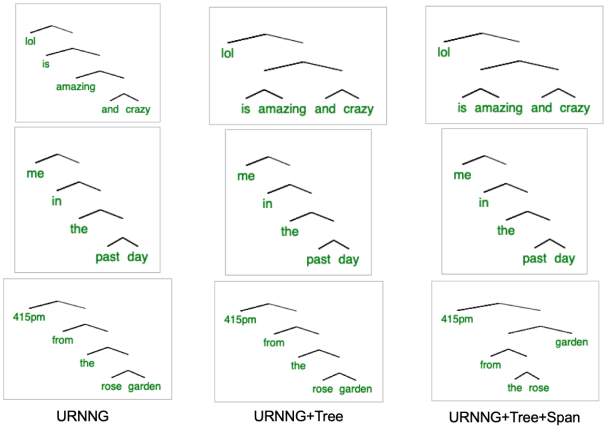


図 6: 5 単語の文における構文解析結果例.

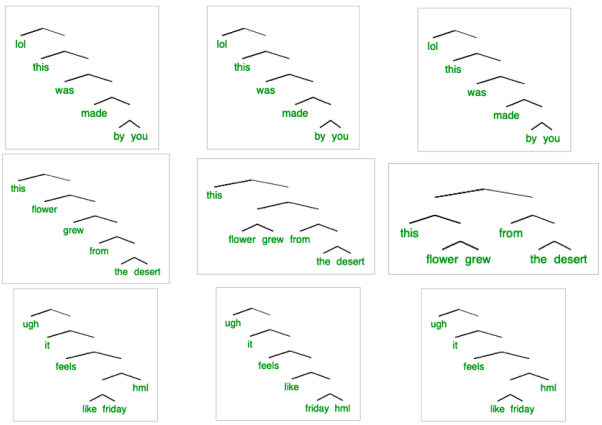


図 7: 6 単語の文における構文解析結果例.

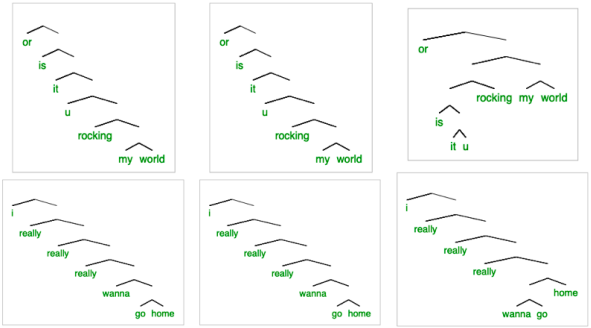


図 8: 7 単語の文における構文解析結果例.

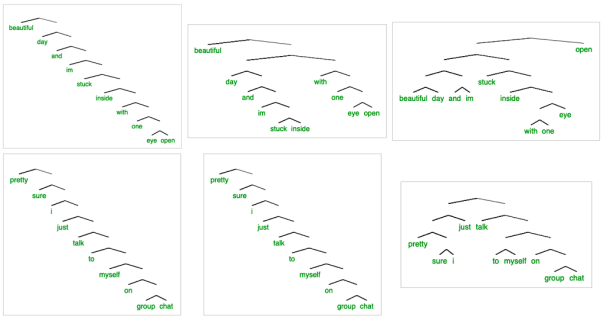


図 9: 10 単語の文における構文解析結果例.

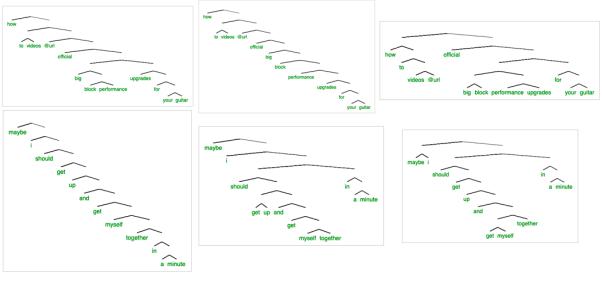


図 10: 12 単語の文における構文解析結果例.