文法変分自己符号化器を用いた統語構造の連続的変換

折口希実
お茶の水女子大学
g1620512@is.ocha.ac.jp

Lis Kanashiro Pereira
お茶の水女子大学
kanashiro.pereira@ocha.ac.jp

小林一郎
お茶の水女子大学
koba@is.ocha.ac.jp

1 はじめに

ニューラル言語モデル (NLM)[1] 及び word2vec[2] は，自然言語処理技術に大きな影響を及ぼし，多くの NLP アプリケーションの基盤構築に利用されている．一方で，それらは言語モデルを上手く表現するが，文の統語構造が十分に反映されているとはいえない．また自然言語を処理するのに用いられる再帰型ニューラルネットワーク (RNN) では，それ自身のアーキテクチュアの制約により，統語情報のような構造化情報を効果的に処理することができない．深層学習のフレームワークで構造情報を処理する方法として，Kusuner ら [3] は離散な構造を持つデータを連続な値，つまり潜在空間における埋め込みベクトルとして扱うことができる，変分自己符号化器 (VAE) モデル [4] に基づいた新しい VAE である Grammar Variational Autoencoder (GVAE) を提案している．本研究では，GVAE モデルを改良し，自然言語文の統語構造を連続的に変換可能にし，ガウス分布によって表された潜在空間からサンプリングする点の変化によって連続的に統語構造を変換可能にする手法を開発する．さらに，潜在空間にエンコードされたベクトルを使用し，統語構造の観点から文の類似性を測定する方法を提案する．本研究で開発する方法は，換言や構文変換、自然言語生成 (NLG) などの様々な自然言語のアプリケーションのための有効な基礎研究になると期待する．

2 関連研究

深層学習を用いて自然言語の統語構造に対するアプローチとして，Shi ら [5] は、エンコーダ-デコーダモデルが構造情報を学習可能かどうかを調べ，原文の多くの構造の詳細が生成されたテキストにおいて十分でなく欠落していることを示した．デコーダは基本的に NLM のフレームワークに基づいて文を生成するため，明示的な構文情報の不足による非文が生成される場合が多くある．この背景を踏まえて，構造に関する知識を活用して文生成の品質を向上させることを目的として，いくつかの研究がこの問題に取り組んできた．Bastings ら [6] は，グラフ畳み込みネットワーク (GCN) を使用して，統語構造を NMT のアテンションに基づくエンコーダ-デコーダモデルに組み込む方法を提案した．Chen ら [7] は，ソース側の構文木を，シーケンシャル表現と木構造表現の両方を学習する双方向ツリーエンコーダーとアテンションメカニズムを用いたソース側の構造が反映される（ツリーカバレッジ）モデルに明示的に組み込むことで，NMT モデルの改善を行った．NLG の研究では，Deriu ら [8] は，生成されたテキストの語彙の変動性と構造上の特徴を制御することにより，より多様な文を生成するための構造操作について提示している．Hu ら [9] は，VAE と敵対的生成ネットワーク (GAN)[10] を使用して，潜在空間での情報を操作することでテキスト生成において構文構造の改善を伴う制御を実現した．本研究のように潜在空間において構造情報を扱う研究として，Bao ら [11] は，分散表現によって意味を扱う潜在空間のみを用いて文を生成することは構造情報を明示的にモデル化していないことを指摘しており，DSS-VAE と呼ばれる意味空間と構造空間のもつれを解消したそれぞれの空間から文を生成する VAE を提案している．

3 文法変分自己符号化器

Kusner ら [3] が提案した自己符号化器 (GVAE) は，入力として離散なデータを扱うことができる VAE モデルの一種であり，文法の生成規則も使用することが可能である．潜在空間のサンプリング点を僅かに変えることで離散情報である自然言語の統語構造をシームレスに変換することができる．彼らはその適用例として，分子の化学構造を ASCII 符号の英数字で文字列化した表記法である SMILES 記法を用いた分子構造に対して連続的な表現および変換方法を示している．

GVAE モデルの概要を図 1 に示す．

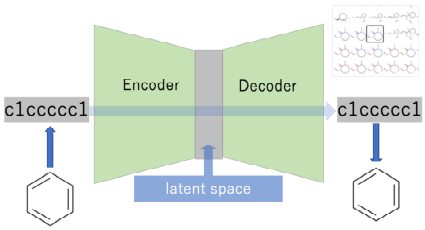

図 1 GVAE の概要．潜在空間でサンプリングされた値に応じて，様々な構造の分子を生成できる．

4 統語構造に対する GVAE

GVAE を適用して自然言語の統語構造の類似性を潜在空間において計測可能としつつ，潜在空間でのサンプリングする値の変化を通じて与えられた文の統語構造の類似関係を制御し統語構造を変換する．GVAE のフレームワークを自然言語の統語構造に適用する場合，Kusner ら [3] で示される SMILES 記法に基づいた分子に適用する場合とは異なり，自然言語文の統語構造は非終端記号の数と文法の生成規則数の両方の点でより複雑となる．実際に，SMILES 記法では終端記号の数は 30，構文規則の数は 76 に対して，本研究で GVAE に用いた自然言語構文を処理するための実験設定では，前者として品詞情報を扱い，後者では短文を表現するための文脈自由文法の規則を採用したことから，それぞれ 27 と 320 となる[1] しかし，これでは生成規則を適用させるための制約もなく，非文となる統語構造を生成する可能性がある．そのためそれを回避するために，文法規則として確率文脈自由文法 (PCFG) を採用する．

4.1 GVAE の処理の流れ

以下，GVAE の処理の流れを説明する (図 2 参照)．

Encoding 図 2 の① から④ で示されるプロセスがエンコーダーによって実行される．① では，PCFG の生成規則が定義されている．② では，自然言語文の統語構造を表す一連の終端記号がシステムに入力され，その入力シーケンスは viterbi parser [2] を使用して① で定義された PCFG を用いて解析される．その後，最も確率的に可能性の高い統語構造に変換され，前順走査によりこの構文木を一連の PCFG の生成規則に分解する．③ では，構文木で使用された規則が抽出され 1-hot ベクトルに変換される．1-hot ベクトルの数は抽出された生成規則の数と同じであり，各ベクトルは用いられた規則を指し示している．④ では，1-hot ベクトルは深層畳み込みニューラルネットワーク (CNN) を介して潜在空間に表される．

Decoding 図 2 の④ から⑥ で示されるプロセスがデコーダによって実行される．⑤ では，まず最初にエンコードされたベクトルが潜在空間からサンプリングされ，次に複数の正規化されていないベクトルが再帰型ニューラルネットワーク (RNN) によって生成される．Last-in First-out (LIFO) スタックを使用することにより，最も可能性の高い PCFG の生成規則を取り出すことができ，⑥ で出力されるように，取り出した規則の終端記号に基づいて統語構造を生成することができる．

Training エンコーダの出力を $\mathbf{X}$ とし，生成規則の総数を K とする．デコーダでは，RNN の時間ステップ t の最大値は T_{max} であり，生成されたベクトルの集合は行列 $\mathbf{F} \in \mathbb{R}^{T_{max} \times K}$ として表す．文法における非終端記号は α とし，生成されたベクトルをマスクするベクトルは $\mathbf{F} \in \mathbb{R}^{T_{max} \times K}$ である．式 (1) で示されている分布は，時間ステップ t でマスクされたベクトルから PCFG の生成規則をサンプリングするために用いられる．また，式 (1) の (t, k) は，行列 $\mathbf{F}$ の (t, k) 要素を指している．$\mathbf{z}$ は GVAE の潜在変数を表し，$m_{\alpha,k}$ は k 番目の PCFG 規則と非終端記号 α の要素のベクトルに対するマスクベクトルを示す．

$$p(\mathbf{x}_t = k|\alpha, \mathbf{z}) = \frac{m_{\alpha,k} \exp(f_{tk})}{\sum_{j=1}^{K} m_{\alpha,k} \exp(f_{tj})}. \quad (1)$$

$q(\mathbf{z}|\mathbf{X})$ はエンコーダーの出力である平均と分散パラメータを持つガウス分布である．損失関数 $\mathcal{L}(\phi, \theta; \mathbf{X})$ を推定するために必要な変分下限 (ELBO) は式 (2) のように計算される．

$$\mathcal{L}(\phi, \theta; \mathbf{X}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{X})}[\log p_\theta(\mathbf{X}, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{X})] \quad (2)$$

4.2 統語構造の類似性

GVAE では，モデルの潜在空間はガウス分布で表されているため，分布からサンプリング値を徐々に変化させることで、入力構造から僅かに異なる統語構造を取得することができる．また，統語構造は潜在変数を反映していることから，統語構造の類似性は潜在変数ベクトル $\mathbf{z}$ を使用することで測定でき

1) 実験設定は，本研究の実験にて短く単純な文を処理するのにおいてはちょうど十分な数であることを予備実験を通じて確認している．
2) https://www.nltk.org/_modules/nltk/parse/viterbi.html

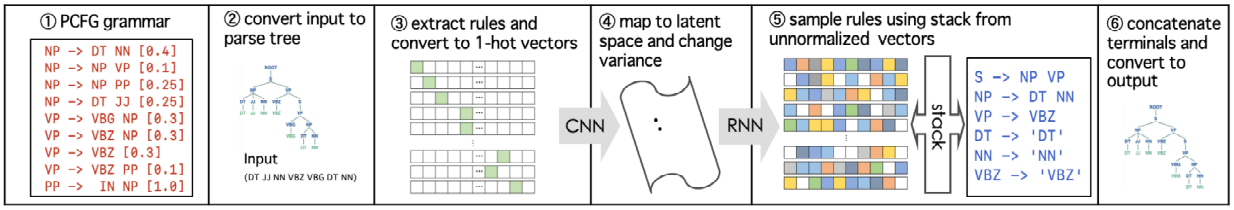

図 2　自然言語の統語構造に対する GVAE のプロセス

表 1　実験設定

データセット	Microsoft COCO
使用文数	6668
PCFG の生成規則数	320
epoch(train) 数	40
バッチサイズ	500
最適化アルゴリズム	Adam
学習率	0.001
損失関数	Binary Cross Entropy (BCE)

る．そのため，cos 類似度を適用してベクトルを計算することで統語構造の類似度を測定する手法としても利用可能となる．

5　実験

潜在空間で表されるガウス分布からのサンプリング点を徐々に変化させることによる統語構造の連続的変換および異なる統語構造の類似性を測定する方法を検証するための実験を行う．

5.1　実験設定

表 1 に実験設定を示す．パラメータは [3] を元にして epoch 数についてはは学習可能な最大値を取りました．本研究では，およそ 10 語程度の文章に絞って行なっており，MicrosoftCOCO データセット[3)]からこの基準を満たす 6,668 の文を選択し，StanfordCoreNLP [4)] を用いて文を解析を行い，GVAE で使用される生成規則を構築した．また，PCFG を構築するために生成規則とその確率を計算して導出し，6668 文の解析に必要となる 320 の生成規則を取得した．

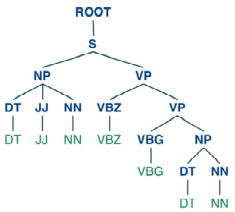

図 3　統語構造 1

5.2　実験結果

統語構造の変換　統語構造 1 ’DT JJ NN VBZ VBG DT NN’ (図 3 参照) を GVAE の入力としている．

3）https://cocodataset.org/home
4）https://stanfordnlp.github.io/CoreNLP/

統語構造 1 の GVAE の出力結果を図 4 に示す．2 次元ガウス分布として表される GVAE の潜在空間のサンプリング点の変化による統語構造の段階的な変化を示している．サンプリング点は、分散軸に従って 0.01 刻みで移動している．図 4 から，統語構造が徐々に変化する様子を視認することができる．

統語構造の類似性　統語構造の類似性を，cos 類似度を用いて潜在変数 **z** で測定した．図 5 は，図 4 の中央に表示されている統語構造と各構造がどの程度類似しているかを示す．下平面の 2 軸の双方が平均値からの変化を指し，縦軸は cos 類似度を指す．表 2 は，統語構造を僅かに変化させていった際の cos 類似度を表している．表 2 における文は統語構造に適当な単語を当てはめて作成した例文である．

表 2　統語構造の違いによる cos 類似度の変化

Sentence examples	統語構造	cos 類似度
A boy is riding a horse.	**DT NN VBZ VBG DT NN**	—
A girl is riding an elephant.	DT NN VBZ VBG DT NN	1
Two boys are riding a horse.	CD NNS VBZ VBG DT NN	0.99999
A boy rides a horse.	DT NN VBZ DT NN	0.99992
A young boy is riding a horse.	DT JJ NN VBZ VBG DT NN	0.99986

5.3　考察

図 3 を入力として潜在変数のサンプリング点を徐々に変えることにより生成した 81 の統語構造が図 4 であり，実際の生成結果に緩やかな構造の変化が見られるかを検証する．サンプリング点が変更されていない場合において図 3 に示されている入力は，図 4 の中央に示されている構造と一致していることが確認できる．図 4 から，潜在変数 **z** の値を徐々に変化させることで統語構造がスムーズに変換されているようにも見える．また，図 5 よりサンプリング点の 2 次元方向が分散に対して同じである場合，構造の類似性が高くなることがわかった．一方で，連続的な変換が起こっていないように見える箇所も確認した．例えば，下から 4 つ目で右から 2 つ目の交差点にある構造は，隣接する統語構造とはかなり異なっているように見えることがわかる．

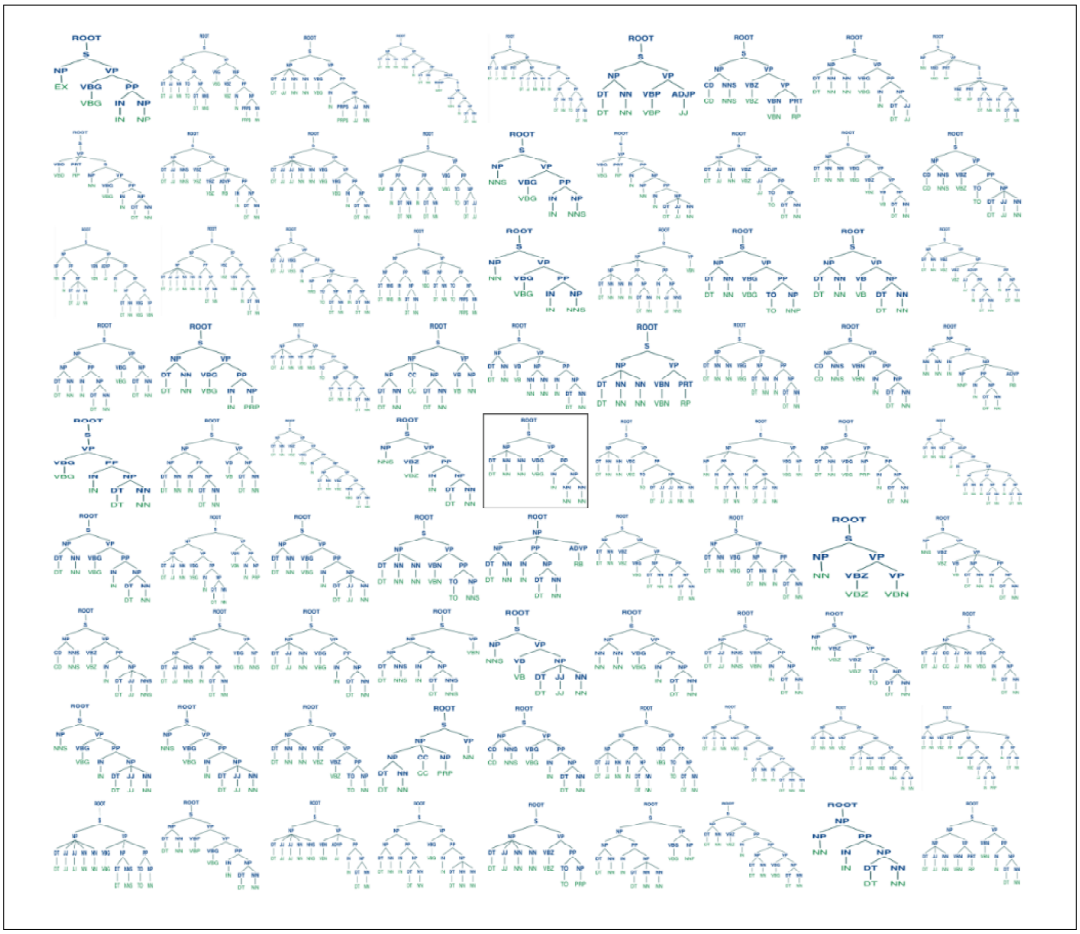

図 4　統語構造 1 の連続的な変換 (サンプリング点は平均から 0.01 刻みで移動)

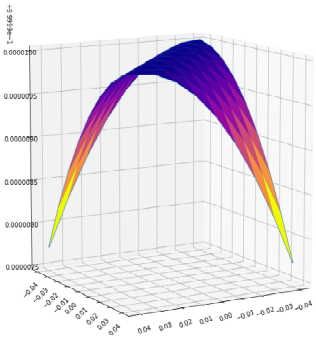

図 5　分散の変化による統語構造間での cos 類似度

また，別の統語構造を入力とした実験も行ったが図 4 と同様な結果を得た．統語構造の連続的変換がうまくいっていない箇所がある原因として，訓練用のデータが不十分であったという可能性が考えられる．[3] におけるモデル訓練データ数は，文法の生成規則や終端記号数の点で自然言語のほうが化学式の構造よりも複雑であるにもかかわらず，本研究の 10 倍であり，よりシームレスな統語構造変換に関してはモデルの訓練に十分なデータを増やす必要がある．また，表 2 から統語構造の変化が大きくなると cos 類似度が低下することが読み取れた．

6　おわりに

本研究では，GVAE を改良し，潜在空間のサンプリング点を徐々に変更することで，自然言語文の統語構造をシームレスに変換できる手法を開発した．実験を通じて，入力文の統語構造から徐々に変化する構造を生成し，統語構造の類似性を潜在空間で表現されるベクトル間の cos 類似度によって測定できることを示した．今後の課題として，統語構造のよりシームレスな変換を実現するために訓練データと文法規則の数を増やすことを考えている．さらに，提案手法を用いることで，複数文間における統語構造の類似性を正しく測定する手法を確立することを考えている．提案手法は，ほぼ同一の意味内容を保ちつつ大きく統語構造を変えない換言や文の類似性の測定，要約などの様々な研究に役立つと期待している．

参考文献

[1] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pp. 932–938. MIT Press, 2001.

[2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 26, pp. 3111–3119. Curran Associates, Inc., 2013.

[3] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. Vol. 70 of *Proceedings of Machine Learning Research*, pp. 1945–1954, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[4] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[5] Xing Shi, Inkit Padhi, and Kevin Knight. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1526–1534, Austin, Texas, November 2016. Association for Computational Linguistics.

[6] Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1957–1967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[7] Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1936–1945, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[8] Jan Milan Deriu and Mark Cieliebak. Syntactic manipulation for generating more diverse and interesting texts. In *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 22–34, Tilburg University, The Netherlands, November 2018. Association for Computational Linguistics.

[9] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. Vol. 70 of *Proceedings of Machine Learning Research*, pp. 1587–1596, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

[11] Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6008–6019, Florence, Italy, July 2019. Association for Computational Linguistics.