# 階層文法言語モデルを用いた言語生成

吉野 幸一郎 [1,3], グエン マイ [1], 高村 大也 [2], 能地 宏 [2], 中村 哲 [1,3]
[1] 奈良先端科学技術大学院大学
[2] 人工知能研究センター (AIRC), 産業技術総合研究所
[3] 革新知能統合研究センター (AIP), 理化学研究所

## 概 要

言語生成において、ドメイン移植性は大きな課題の一つである。本研究では、品詞系列からなる文法情報の同時予測を行うことで、ドメイン移植性の高い言語生成器の構築を行う。パープレキシティを用いた言語モデル評価と、生成文に対する主観評価実験を行った結果、少ないデータで適応可能な言語生成器を構築できたことを確認した。

## 1 はじめに

言語生成とは、自然言語文とこれに解釈を与える意図のペアが与えられたとき、意図を入力として対応する自然言語文を生成する課題である。古くはルールやテンプレートを用いた生成が行われていたが [1, 2, 3]、近年ではニューラルネットワークに基づく確率的言語モデルをデコーダに用いた言語生成を行うことが一般的となりつつある [4, 5]。

確率的言語モデルは、これまでの生成単語を入力として次の最尤単語を求める確率モデルであり [6]、ニューラル言語モデルではこの予測にニューラルネットワークを用いる。この際、ニューラルネットワークに再帰構造を持たせることで、直前だけでなく過去の単語系列を考慮した最尤単語を求めることが一般的である [4, 5, 7]。このニューラル言語モデルを言語生成のタスクに利用する場合、デコーダは単語予測を逐次的に行う確率的言語モデルとし、対応する発話意図をエンコーダに入力することで対応学習を行う。

ニューラル言語生成モデルは単語系列を逐次予測するため、学習に用いたデータのドメインに強い影響を受ける。そこで、学習データとテストデータのドメインが異なる場合、学習されたモデルをテストデータのドメインに対して無情報、あるいは少ない情報で適応することが必要となる [8]。

こうした言語モデルの適応は、適応前のモデルから十分に予測できる単語とそうではない単語に分けることができる。例えば、固有名詞などは適応が必要であるのに対し、助詞などは様々なドメイン間で共通する場合が多い。そこで本研究では、品詞系列がドメイン移植性の高い汎化情報を持つことを仮定し、階層文法言語モデルを用いた言語生成システムを提案する。

## 2 階層文法言語モデルと言語生成

### 2.1 言語モデルの定式化

確率的言語モデルは単語系列 $X_n = x_1, x_2, ..., x_n$ において、ある単語 $x_i$ をそれ以前の部分系列 $X_{i-1}$ から予測する確率モデル $P(x_i|X_{i-1})$ である。再帰構造を持つニューラルネットワークを利用する場合、各単語は以下のネットワークによって予測する。

$$\mathbf{h}_i = \sigma(\mathrm{emb}(x_{i-1})W_{xh} + \mathbf{h}_{i-1}W_{hh} + \mathbf{b}_h) \quad (1)$$
$$\mathbf{x}_i = \mathrm{softmax}(\mathbf{h}_i W_{hx} + \mathbf{b}_x) \quad (2)$$
$$x_i = \mathrm{argmax}_x \mathbf{x}_i \quad (3)$$

ここで $\sigma$ は活性化関数であり、softmax は実数値ベクトルを確率ベクトルに変換する関数である。$W_{xh}$、$W_{hh}$、$W_{hx}$ はそれぞれ入力層、隠れ層、出力層で学習するパラメータ行列である。$x_i$ は予測単語であり、softmax によって得られる予測単語の確率変数 $\mathbf{x}_i$ から argmax によって最尤の予測単語 $x_i$ を決定する。なお、以降ではこの argmax の操作は省略する。決定された単語 $x_i$ を次の隠れ層に入力する際には、関数 emb によって分散表現 [9] へと変換する。$x_i$ の予測に、直前の予測単語 $x_{i-1}$ とそれ以前の単語予測に用いた隠れ層 $\mathbf{h}_{i-1}$ を用いることで、再帰的にこれまでの単語系列 $X_{i-1}$ の情報を用いる。

### 2.2 階層文法言語モデル

一般的な言語モデルは直前の単語を入力として次の単語を予測するため、学習とテストのドメインが大きく異なるような場合に不適合が起こる。そこで各単語をクラス化することによってこの不適合を埋めることを考える。本研究では単語系列を品詞系列に変換し、この情報を同時に用いることでドメイン不適合の問題を軽減する。このような手法はマルチタスク学習 [10] として知られ、それぞれの予測が相互に精度向上をもたらすことを期待する。今回は単語 $x_i$ が属する品詞クラスを $z_i$ としたとき、これらの同時確率を

$$P(x_i, z_i|X_{i-1}, Z_{i-1})$$
$$= P(x_i|z_i, X_{i-1}, Z_{i-1})P(z_i|X_{i-1}, Z_{i-1}) \quad (4)$$
$$\approx P(x_i|X_{i-1}, Z_{i-1})P(z_i|X_{i-1}, Z_{i-1}) \quad (5)$$

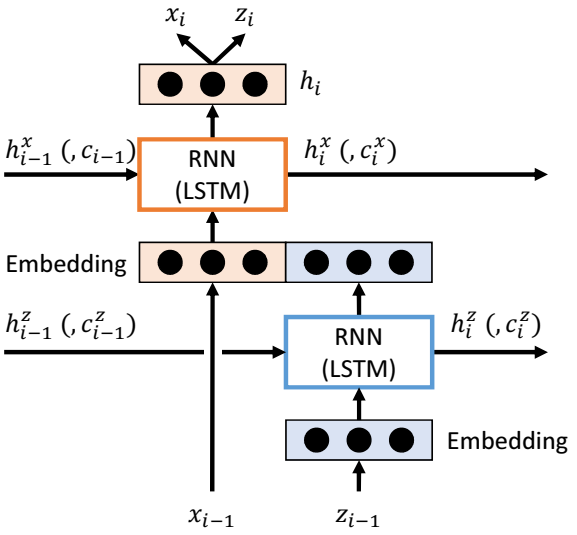


図 1: マルチタスクモデル

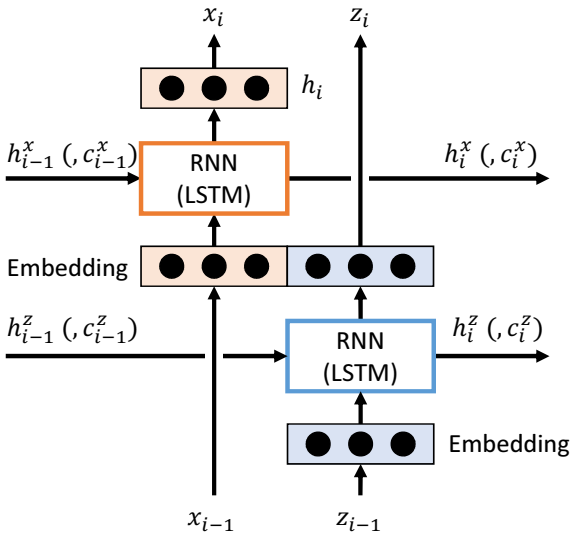


図 2: カスケードモデル

の近似によって求める。このモデルをマルチタスクモデルと呼ぶ。また、クラス言語モデル [6] のように先に品詞情報を予測し、これを該当する単語の予測に役立てる方法も考えられる。そこで式 (4) を

$$(4) \approx P(x_i|z_i, X_{i-1}, Z_{i-1})P(z_i|Z_{i-1}) \quad (6)$$

によって近似するモデルも検討する。このモデルをカスケードモデルと呼ぶ。

それぞれのモデルが持つネットワーク構造について図 1、図 2 に示し、以下の節で詳細を説明する。なお、ニューラルネットワークとして Recurrent neural networks (RNN) を用いる場合、各時点で引き継がれるのは隠れ層 $\mathbf{h}_i$ のみとなるが、Long short-term memory neural networks (LSTM) を用いる場合はセル状態 $c_i$ もあわせて引き継ぐ。以降では簡単のため RNN の場合を説明する。

### 2.2.1 マルチタスクモデル

マルチタスクモデルにおいては、まず品詞情報を入力とする RNN で隠れ層を求める。

$$\mathbf{h}^z_i = \sigma(\mathrm{emb}(z_{i-1})W^z_{zh} + \mathbf{h}^z_{i-1}W^z_{hh} + \mathbf{b}^z_h) \quad (7)$$

この隠れ層を、単語情報を入力とする別の RNN の入力層に連結し、新たに求めた隠れ層から品詞 $z_i$ および単語 $x_i$ を求める。

$$\mathbf{h}^x_i = \sigma((\mathrm{emb}(x_{i-1}) \odot \mathbf{h}^z_i)W^x_{(x+z)h} + \mathbf{h}^x_{i-1}W^x_{hh} + \mathbf{b}^x_h) \quad (8)$$

$$\mathbf{z}_i = \mathrm{softmax}(\mathbf{h}^x_i W^x_{hz} + \mathbf{b}^z_z) \quad (9)$$

$$\mathbf{x}_i = \mathrm{softmax}(\mathbf{h}^x_i W^x_{hx} + \mathbf{b}^x_x) \quad (10)$$

式 (8) 中の $\mathbf{h}^z_i$ は式 (7) で求めた品詞予測を行う RNN の隠れ層であり、単語 $x_i$ を分散表現に変換したベクトルと連結することで RNN への入力とする。

このモデルでは、単語系列、品詞系列双方の情報が単語 $x_i$ と品詞 $z_i$ の決定に寄与するため、それぞれの情報が精度向上に寄与することが期待される。一方で、品詞予測と単語予測が同時に行われるため、それぞれの予測が独立に行われる可能性もある。

### 2.2.2 カスケードモデル

カスケードモデルにおいては、まず直前の品詞 $z_{i-1}$ から次の単語の品詞 $z_i$ を予測する以下のモデルを構築する。

$$\mathbf{h}^z_i = \sigma(\mathrm{emb}(z_{i-1})W^z_{zh} + \mathbf{h}^z_{i-1}W^z_{hh} + \mathbf{b}^z_h) \quad (11)$$

$$\mathbf{z}_i = \mathrm{softmax}(\mathbf{h}^z_i W^z_{hz} + \mathbf{b}^z_z) \quad (12)$$

ここでマルチタスクモデルとの異なりは、$z_i$ の予測に $X_{i-1}$ が寄与しない点である。また、次の単語 $x_i$ を求めるモデルは以下のように構築する。

$$\mathbf{h}^x_i = \sigma((\mathrm{emb}(x_{i-1}) \odot \mathbf{h}^z_i)W^x_{(x+z)h} + \mathbf{h}^x_{i-1}W^x_{hh} + \mathbf{b}^x_h) \quad (13)$$

$$\mathbf{x}_i = \mathrm{softmax}(\mathbf{h}^x_i W^x_{hx} + \mathbf{b}^x_x) \quad (14)$$

このモデルでは、品詞の予測を一度行った上でこの情報を単語予測に用いるため、より品詞の情報が単語予測に反映されることが期待される。

### 2.2.3 損失関数

これまで説明した 2 つのモデルは、いずれも出力として単語、品詞の双方を持つため、それぞれに対する softmax cross entropy loss $L_x$ と $L_z$ によって

$$L_{all} = L_x + \lambda L_z \quad (15)$$

として損失関数を定義する。$\lambda$ は結合重みであり、今回は $\lambda$ =1.0 に設定した。

### 2.3 意図の入力

これまでに説明した言語モデルはデコーダとして用いる。意図を入力として対応する自然言語文を生成しようとする場合、意図をネットワークに入力としてデコーダへの入力 $\mathbf{h}_0$ としなければならない。意図の入力方法としては n-hot 表現とする手法 [4] や意図のセットを系列として入力する手法 [5] などが提案されているが、ここでは系列として意図を入力する方法を取る。

意図は一般に key-value $k:v$ の集合として定義する。例えばレストラン案内の場合、{area:河原町, price:high, number:2} のように記述する。これを系列 $F$ =area, 河原町, price, high, number, 2 のように変形し、各要素 $f_i$ を前から順番に RNN を用いたエンコーダへ入力する。エンコーダは以下の式で定義される。

$$\mathbf{h}_i = f_i W^e_{fh} + \mathbf{h}_{i-1} W^e_{hh} + \mathbf{b}^e_h \qquad (16)$$

各言語モデルを用いたデコーディング時は、これらの入力に対応する各隠れ層に対して注意機構 [11] を用いてデコーダの隠れ層および出力層への対応を取る。

## 3 実験

本研究では言語生成のデコーダ部分を担う言語モデルの改善を提案したため、まず言語モデルの評価を行った。さらに、実際にそれらの機構が言語生成の改善に寄与したかを確認する実験を行った。

### 3.1 言語モデルとドメイン移植性の評価

提案した 2 種類の言語モデルの性能を確認するため、テストセットパープレキシティ[12] を用いた評価を行った。本タスクではエンコーダ部分は用いず、デコーダ部分によって次単語を予測する言語モデルタスクを行った。比較手法としては、2 層の LSTM による言語モデルを用いた。学習およびテストには、Penn Treebank (PTB)[13] と WikiText2[14] を用いた。なお、品詞の付与には Stanford Core NLP toolkit [15] を用い、学習/開発/テストの分割は先行研究に従った [16]。実験にあたり、大文字は小文字化し、数詞は数詞シンボルに置き換えた。また PTB の語彙は頻度上位から 10,000 件、WikiText2 の語彙は頻度上位から 30,000 件を用い、その他の単語は未知語シンボルに置き換えた。開発/テストデータにおいても学習データの語彙を適用し、それ以外は未知語とした。先行研究にならい、隠れ層のユニット数は 1,500 とした。また Optimizer には SGD を用いた。

まず、各コーパスでのテストセットパープレキシティの値を表 1 に示す。この結果から、マルチタスク、カスケードのいずれも言語モデルの改善に有効であるものの、どちらがよいかについては議論の余地があることがわかる。

| | PTB | | WikiText | |
|---|---|---|---|---|
| | Valid | Test | Valid | Test |
| Baseline | 75.27 | 75.58 | 93.68 | 88.98 |
| Multitask | **68.19** | **68.36** | 81.33 | 76.49 |
| Cascade | 68.28 | 68.63 | **80.91** | **75.98** |

表 1: 言語モデルタスクにおけるパープレキシティ

| | Valid | Test |
|---|---|---|
| Baseline | 60.60 | 60.72 |
| Multitask | 56.04 | 56.16 |
| Cascade | 56.86 | 56.98 |

表 2: ドメイン移植時のパープレキシティ

次に、WikiText で事前学習したモデルに対して PTB の学習データで fine-tuning したものを用いて、PTB のテストセットパープレキシティを求めたものを表 2 に示す。ここで、提案法、ベースラインはいずれも WikiText2 で事前学習し、PTB で fine-tuing したものである。また、fine-tuing に用いるデータサイズとテストセットパープレキシティの関係を図 3 に示す。この結果から、提案する階層的文法言語モデルは、適応対象のドメインデータが十分に存在しない場合に高い効果を発揮することが見て取れる。

### 3.2 言語生成の評価

続いて、提案する言語モデルを用いた言語生成の評価を行った。言語生成の自動評価指標は様々なものが提案されているが、人手評価との相関が高いと結論されたものが存在しないことから、自動評価と人手評価の両方を行った。自動評価には BLEU[17]、NIST[18]、ROUGE-L[19] を用いた。人手評価は 5 段階の主観評価で、入力された意図に対応する情報が出力されているか（有益性）、自然な文が生成されているか（自然性）を付与してもらった。本実験には、我々が構築した京都観光案内に関する言語生成データセット [20] を用いた。本データセットでは 283 通りの意図表現に対して 3,296 発話をクラウドソーシングを用いて収集した。このデータを学習/開発/テストに 2,800/200/296 発話ずつ分割して利用した。人手評価は各文 3 名を評価に割り当て、スコアの平均を取った。

2.3 節で述べたエンコーダに対して、LSTM (Seq2seq)、マルチタスク、カスケードのそれぞれのデコーダを接続したモデルを比較に用いた。これらのネットワークではいずれも注意機構を加え、マルチタスク、カスケードのモデルではそれぞれ単語のデコード時にのみ注意機構を利用した。最適化には SGD を用い、エンコーダの隠れ層のサイズは 256、デコーダの隠れ層のサイズは 512、分散表現のベクトルサイズは 256 とした。また、これ以外のベースラインとして Semantically Conditioned LSTM (SC-LSTM)[4] を

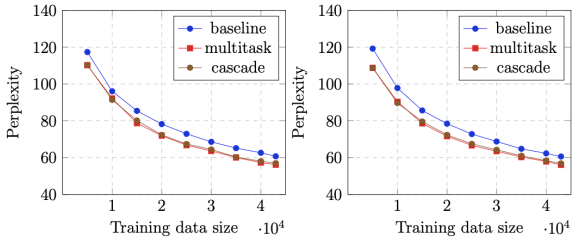


図 3: データサイズとパープレキシティ

| | BLEU | NIST | ROUGE |
|---|---|---|---|
| SC-LSTM | 0.43 | 6.03 | 0.64 |
| Seq2seq | 0.44 | 6.04 | 0.65 |
| Multitask | 0.46 | 6.09 | 0.63 |
| Cascade | 0.45 | 6.12 | 0.67 |

表 3: 自動評価結果

用いた生成も行った。自動評価による比較を表 3 に、人手評価による比較を表 4 に示す。

これらの評価の結果、自動評価においても人手評価においても、提案したマルチタスク、カスケードのモデルはわずかな改善が見られるものの、有意な改善は見られなかった。個別の評価指標を見ると BLEU や NIST、自然性といった文の流暢性が関わるような項目でわずかな改善が見られる。これは、言語モデルの評価においてパープレキシティが向上したことと一致する。

## 4 おわりに

本研究ではドメイン移植性の高い言語モデルと言語生成を実現するため、階層的文法言語モデルを用いた言語生成器を提案した。マルチタスクとカスケードの 2 つのモデルを提案し、言語モデルのドメイン適応評価と言語生成の評価では一般的な LSTM 言語モデルからの改善を確認した。今後は言語生成においても移植性の評価を行う。また、E2E NLG Challenge [21] などの一般的なデータセットでの生成評価も行う。

| | 有用性 | 自然性 |
|---|---|---|
| SC-LSTM | 4.28 | 4.32 |
| Seq2seq | 4.30 | 4.36 |
| Multitask | 4.26 | 4.38 |
| Cascade | 4.29 | 4.37 |

表 4: 人手評価結果

## 参考文献


[1] Gabor Angeli, Percy Liang, and Dan Klein. A simple domain-independent probabilistic approach to generation. In *Proc. EMNLP*, pp. 502–512, 2010.

[2] Ravi Kondadadi, Blake Howald, and Frank Schilder. A statistical nlg framework for aggregated planning and realization. In *Proc. ACL*, pp. 1406–1415, 2013.

[3] 山崎健史, 吉野幸一郎, 前田浩邦, 笹田鉄郎, 橋本敦史, 舩冨卓哉, 山肩洋子, 森信介ほか. フローグラフからの手順書の生成. 情報処理学会論文誌, Vol. 57, No. 3, pp. 849–862, 2016.

[4] Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proc. EMNLP*, pp. 1711–1721, 2015.

[5] Ondřej Dušek and Filip Jurcicek. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proc. ACL*, pp. 45–51, 2016.

[6] 森信介, 西村雅史, 伊東伸泰ほか. クラスに基づく言語モデルのための単語クラスタリング. 情報処理学会論文誌, Vol. 38, No. 11, pp. 2200–2208, 1997.

[7] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Proc. INTERSPEECH*, 2010.

[8] Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. Cross-lingual word embeddings for low-resource language modeling. In *Proc. EACL*, pp. 937–947, 2017.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. NeurIPS*, pp. 3111–3119, 2013.

[10] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. ICML*, pp. 160–167, 2008.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*, 2015.

[12] Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, Vol. 62, No. S1, pp. S63–S63, 1977.

[13] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993.

[14] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[15] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proc. ACL*, pp. 55–60, 2014.

[16] Tomáš Mikolov, Anoop Deoras, Stefan Kombrink, Lukáš Burget, and Jan Černockỳ. Empirical evaluation and combination of advanced language modeling techniques. In *Proc. INTERSPEECH*, 2011.

[17] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. ACL*, pp. 311–318, 2002.

[18] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proc. ACL*, p. 605, 2004.

[19] A package for automatic evaluation of summaries. In *Proc. Text summarization branches out at ACL*, 2004.

[20] Captioning events in tourist spots by neural language generation. 情処研報 SIG-NL-240, 2019.

[21] Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. *Computer Speech & Language*, Vol. 59, pp. 123–156, 2020.