忠実な Data-to-Text 生成のための自信度付きデコーダー

Ran Tian* Shashi Narayan Thibault Sellam Ankur P. Parikh
Google Research
{tianran, shashinarayan, tsellam, aparikh}@google.com

1 はじめに

ニューラルな条件付き文生成モデルは，生成テンプレートを明示的に作らずに流暢な文を生成できるという大きな利点を持つが，制御が難しく入力データの意味に忠実でない文を生成してしまう場合がある [7,9,17,18]．これは，生成文の精度が重要視される多くの応用場面において課題となる．

例えば表1では，英語ウィキペディアの人物に関するインフォボックスからバイオグラフィーを生成する WikiBio データセット [8] における一例を示した．まず，生成の目標となるリファレンス文には *bonanno crime family* や *informant* のような，インフォボックスにない情報を含んでいることに注目されたい．このような入力と出力の食い違いは，ニューラルな文生成モデルを訓練するために必要な大規模データセットによく現れる現象であり [5,18]，モデルに誤った生成を強いてしまう恐れがある．実際，ベースラインのコピー機能付き Seq2Seq モデル [13] は，この例に対して *criminal defense attorney* という，インフォボックスにない誤った情報を生成した（インフォボックスにある *FBI* という単語に影響されたかもしれない）．このように誤った生成をする恐れのあるモデルを，新聞記事など情報の信憑性が非常に大事な分野に適用すると深刻な問題になりかねない．

この問題に対して本研究は，自信度付きの新しいデコーダーを提案する．自信度はテスト時に較正（calibration）[2] という手法を通じてより入力データの意味に忠実な文生成を促し，訓練時にはリファレンスの食い違いや学習の難しそうな単語を無視できるようモデルを制御する．同時に，このような自信度はある変分ベイズ目的関数を通して学習される．WikiBio データセットを使った実験で，我々の手法は PARENT Precision [5] および人手評価の忠実性に関する指標において既存研究を大きく上回ったこと示す．例えば表1に，提案モデルは職業を省略し正しい文を生成した．

Wikipedia Infobox	
Frank Lino	
Caption	FBI surveillance photo
Birth date	October 30, 1938
Birth place	Gravesend, Brooklyn, New York, United States

Reference: *Frank "Curly" Lino (born October 30, 1938 Brooklyn) is a Sicilian-American Caporegime in the Bonanno crime family who later became an informant.*

Baseline: *Frank Lino (born October 30, 1938 in Brooklyn, New York, United States) is an American criminal defense attorney.*

Our model: *Frank Lino (born October 30, 1938 in Brooklyn, New York, United States) is an American.*

表1 WikiBio データセットにおける生成例．

2 背景

本研究のベースとなる，エンコーダー・デコーダーモデル [1,14] について概要を述べる．$\boldsymbol{x} = x_1 x_2 \dots x_S$ を長さ S の入力文字列，$\boldsymbol{y} = y_1 y_2 \dots y_T$ を長さ T の出力文字列とする．我々は条件付き確率 $P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{T} P(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x})$ をモデリングする．ここで $\boldsymbol{y}_{<t} = y_1 \dots y_{t-1}$ は $\boldsymbol{y}$ の最初から $(t-1)$-番目までのトークンを含んだ部分列を表す．入力文字列はニューラルネットワーク **enc** によって符号化され，$\boldsymbol{s}_1, ..., \boldsymbol{s}_S = \mathbf{enc}(x_1, ..., x_S)$ とする．

単語 x の埋め込みを $\boldsymbol{e}_x \in \mathbb{R}^d$ とすると，生成確率は

$$P\left(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}\right) = \frac{\exp\left(\boldsymbol{v}_t^\top \boldsymbol{e}_{y_t}\right)}{\sum_y (\exp \boldsymbol{v}_t^\top \boldsymbol{e}_y)}$$

で与えられる．ここで，位置 t における文脈ベクトル

* Google NYC の Visiting Faculty として行った研究．

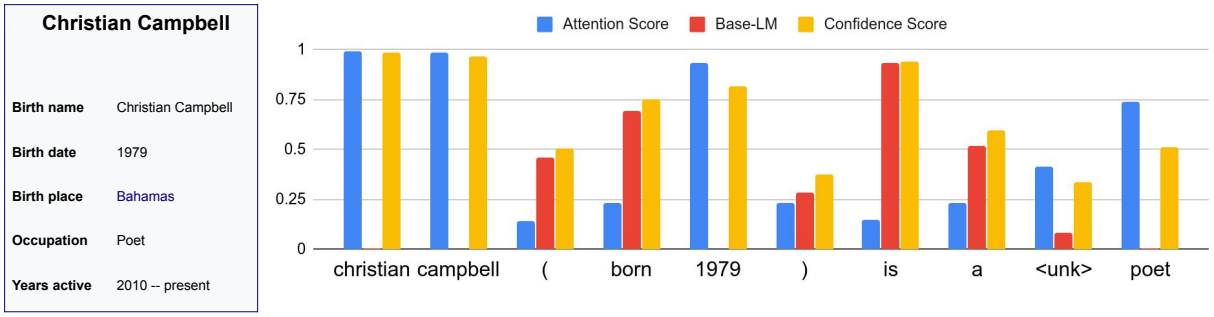

図1　学習された注意度，基準確率，そして自信度の例．内容語に対して基準確率が低く，自信度はより注意度に依存する．

$\boldsymbol{v}_t$ は次のように定義される：

$$\boldsymbol{v}_t = \boldsymbol{a}_t + \boldsymbol{h}_t = \sum_{s=1}^{S} \alpha_{s,t} \boldsymbol{s}_s + \boldsymbol{h}_t$$

この式の右辺一番目の項は注意力ベクトルで，本研究では Luong 型 [11] を使う（式1）．二番目の項は隠れ状態ベクトルで RNN によって与えられる（式2）：

$$\alpha_{s,t} = \frac{\exp(\boldsymbol{s}_s^\top \boldsymbol{W} \boldsymbol{h}_t)}{\sum_{s'} \exp(\boldsymbol{s}_{s'}^\top \boldsymbol{W} \boldsymbol{h}_t)} \tag{1}$$

$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{h}_{t-1}, [\boldsymbol{e}_{y_{t-1}}, \boldsymbol{a}_{t-1}]) \tag{2}$$

ここで $[\cdot]$ はベクトルの連結を表す．デコーダーを RNN に限定したのは，本研究でこれから定義する自信度は生成済みのトークンに依存し，一つずつ計算していく必要があるからである．

コピー機能も考慮する場合，生成確率はコピー確率と混合され，次のようになる [6,13]：

$$\tilde{P}(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) = p_t^{\mathrm{gen}} P(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) + (1 - p_t^{\mathrm{gen}}) \sum_{s:x_s=y_t} \beta_{s,t}$$

ここで p_t^{gen} は位置 t におけるトークンが辞書から生成される確率，$\beta_{s,t}$ はコピーの注意力重みである．単語 x_s が y_t と同じような位置 s において和が取られる．

3　自信度付きデコーダー

複雑なエンコーダー・デコーダーをいかに制御して，より忠実な文生成に導けるだろうか？表1の例では，インフォボックスに通常存在する *Occupation* というフィールドがないにもかかわらず，それを幻視したようにベースラインモデルは誤った生成をした．このように，通常存在するフィールドがない時，忠実でない生成がよく起こる．我々の観察では，このような場合でも，誤ったトークンの生成確率が高く（> 0.5），単に生成確率からでは忠実性を判断するのが難しい．一方，フィールドの欠損は入力データに対する注意力機構から割り出せる可能性があり，実際我々は**注意度**（Attention score）A_t を次のように定義したところ，

$$A_t := \frac{\|\boldsymbol{a}_t\|}{\|\boldsymbol{a}_t\| + \|\boldsymbol{h}_t\|}$$

A_t は幻視トークンに対して低下することを予備実験で確認した．更にコピー機能を使って注意度を次のように修正すると：

$$\tilde{A}_t := p_t^{\mathrm{gen}} A_t + (1 - p_t^{\mathrm{gen}})$$

図1に示すように $\tilde{A}_t$ はコピーしたトークン（e.g. *Campbell*）に対して高く（$\sim$0.9），入力データから生成した場合（e.g. *<unk>*）に中程度（$\sim$0.4），そして機能語やテンプレート要素，またはフィールド欠損の場合に対して低い（$\sim$0.2）ことが観察された．

よって，注意度は幻視トークンと内容語の区別ができるが，機能語やテンプレート要素との区別がまだできない．そこで我々は，基準の言語モデル（Base-LM）の生成確率 $P(y_t|\boldsymbol{y}_{<t})$（**基準確率**と呼ぶ）との比較を取る手法を考えた．基準言語モデルは，入力データの情報にアクセスできない．したがって，もし基準確率が条件付き生成確率と同程度ならば，そのトークンは機能語やテンプレート要素のような，あまり入力データの情報を保持しないトークンと思われる．これより**自信度**（Confidence score）を，条件付き生成確率と基準確率の補間として次のように定義する：

$$C(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) := \tilde{A}_t P(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}) + (1 - \tilde{A}_t) P(y_t \mid \boldsymbol{y}_{<t})$$

機能語やテンプレート要素に対して，$P(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x})$ と $P(y_t|\boldsymbol{y}_{<t})$ は両方高く，すると自信度は注意度によらず

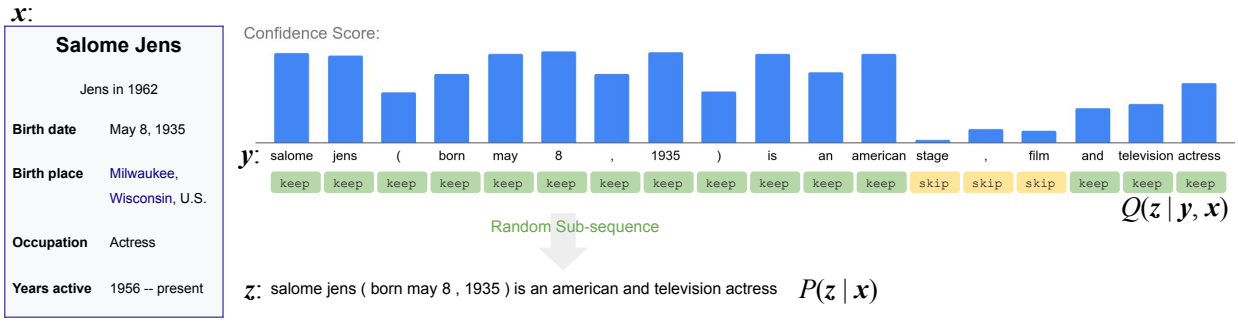

図2　自信度による部分列サンプリングの例.

高いことになる．一方，入力データの情報を保持する内容語に対しては $P(y_t|\boldsymbol{y}_{<t},\boldsymbol{x})$ が $P(y_t|\boldsymbol{y}_{<t})$ より高く，自信度は注意度に大きく依存する（図1）．

3.1　較正（Calibration）

より忠実な文生成を行うために，自信度を使った生成確率の較正（Calibration）[2] を次のように行う：

$$\hat{P}^{\kappa}(y_t|\boldsymbol{y}_{<t},\boldsymbol{x}) := \frac{\mathrm{SG}(C(y_t|\boldsymbol{y}_{<t},\boldsymbol{x}))^{\kappa}\mathrm{SG}(P(y_t|\boldsymbol{y}_{<t},\boldsymbol{x}))}{\sum_w \mathrm{SG}(C(w|\boldsymbol{y}_{<t},\boldsymbol{x}))^{\kappa}\mathrm{SG}(P(w|\boldsymbol{y}_{<t},\boldsymbol{x}))}$$

ここで，$\hat{P}^{\kappa}(y_t|\boldsymbol{y}_{<t},\boldsymbol{x})$ は κ をパラメータとする確率分布の族で，SG（Stop-Gradient）は誤差の逆伝播をブロックし，κ だけが訓練される．訓練では，$\hat{P}^{\kappa}(y_t|\boldsymbol{y}_{<t},\boldsymbol{x})$ と $P(y_t|\boldsymbol{y}_{<t},\boldsymbol{x})$ と $P(y_t|\boldsymbol{y}_{<t})$ の負対数尤度は合同で最小化される：

$$\mathcal{L}_{\mathrm{joint}}(\boldsymbol{y},\boldsymbol{x}) := \sum_{t=1}^{T}\big(-\log\hat{P}^{\kappa}(y_t|\boldsymbol{y}_{<t},\boldsymbol{x}) - \log P(y_t|\boldsymbol{y}_{<t},\boldsymbol{x}) - \log P(y_t|\boldsymbol{y}_{<t})\big)$$

確率分布の族 $\hat{P}^{\kappa}(y_t|\boldsymbol{y}_{<t},\boldsymbol{x})$ は $P(y_t|\boldsymbol{y}_{<t},\boldsymbol{x})$ を含んでいるため(i.e. $\kappa=0$ の時),学習された $\hat{P}^{\kappa}(y_t|\boldsymbol{y}_{<t},\boldsymbol{x})$ の訓練データにおけるパープレキシティは $P(y_t|\boldsymbol{y}_{<t},\boldsymbol{x})$ より高くなることはない．実際，κ は 0 で初期化され正の値に収束する．よって，文生成の自信度が高くなるよう調整され，しかも訓練パープレキシティを犠牲にすることはない．

3.2　変分ベイズによる訓練

文生成の訓練データは，多くの場合ノイジーで外れ値を持つ．これら外れ値はモデルに間違った共起関係を学習させ，誤った生成を引き起こす恐れがある．外れ値の影響を軽減し，より正確なデータでモデルを訓練するために，本研究は自信度の低いトークンを無視できるような訓練手法を提案する．しかし自信度は訓練で使われると同時に学習されなければならないので，変分ベイズ目的関数でこの問題を定式化する．

具体的に，リファレンス文字列 $\boldsymbol{y}=y_1y_2\ldots y_T$ に対して，その信頼できる隠れ部分列 $\boldsymbol{z}=z_1z_2\ldots z_R=y_{\iota(1)}y_{\iota(2)}\ldots y_{\iota(R)}$ を考える．ここで R が部分列の長さ，$\iota:|R|\to|T|$ が包含写像とする．部分列 $\boldsymbol{z}$ の生成確率を最大化したい:

$$P(\boldsymbol{z}\mid\boldsymbol{x}) = \prod_{r=1}^{R}\hat{P}^{\kappa}(z_r\mid\boldsymbol{z}_{<r},\boldsymbol{x})$$

ベイズ法則で $P(\boldsymbol{z}|\boldsymbol{x})$ を $P(\boldsymbol{y}|\boldsymbol{x})$ に関連付けると：

$$P(\boldsymbol{y}\mid\boldsymbol{x}) = \frac{P(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x})\,P(\boldsymbol{z}|\boldsymbol{x})}{P(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})}.$$

を得る．簡単のため，リファレンス文は各訓練事例に対して唯一定まるとして $P(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x})=1$ とする．そして，部分列 $\boldsymbol{z}$ を $\boldsymbol{y}$ に対する “`keep`/`skip`” の系列ラベリングとみなし，自信度を使った次の確率分布で部分列をサンプリングする（図2）：

$$Q(\boldsymbol{z}\mid\boldsymbol{y},\boldsymbol{x}) = \prod_{t=1}^{T}Q_t$$

$$Q_t := \begin{cases} Q_t(\texttt{keep}) \propto C(y_t|\boldsymbol{z}_{\iota(s)<t},\boldsymbol{x})^{\rho} & \text{For } y_t\in\boldsymbol{z} \\ Q_t(\texttt{skip}) \propto \gamma^{\rho} & \text{For } y_t\notin\boldsymbol{z} \end{cases}$$

ここで ρ と γ はハイパーパラメータである．$Q(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})$ を使って事後分布 $P(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})$ を近似すると，不等式

$$-\log P(\boldsymbol{y}\mid\boldsymbol{x}) \le \mathbb{E}_{Q(\boldsymbol{z}|\boldsymbol{y},\boldsymbol{x})}\Big[\log Q(\boldsymbol{z}\mid\boldsymbol{y},\boldsymbol{x}) - \log P(\boldsymbol{z}\mid\boldsymbol{x})\Big]$$

を得る．変分ベイズ目的関数は，この不等式の右側を最小化する．本研究で，部分列は自信度だけによってサンプリングされ，文としての流暢性を考慮しなかったが，驚くことに流暢な文が生成される．

Model	Automatic Evaluation			Human Evaluation		
	BLEU	PARENT (Precision / Recall / F_1)	Avg Len.	Faithful %	Avg Cov.	Fluency %
BERT-to-BERT	44.83	77.62 / 43.00 / 53.13	20.9	77.6	4.33	**98.5** / 99.4
Structure-Aware Seq2Seq	**45.36**	73.98 / **44.02** / 52.81	23.1	66.1	**4.47**	88.6 / 99.7
Pointer-Generator	41.07	77.59 / 42.12 / 52.10	19.1	80.3	4.24	93.1 / 96.0
Confident BERT-to-RNN	33.30	77.98 / 37.21 / 47.90	16.6	85.2*	3.90	92.3 / 94.1
Confident Pointer-Generator	38.10	**79.52** / 40.60 / 51.38	17.0	**86.8***	4.05	95.4 / 96.3
+threshold=0.125	36.62	**80.15** / 39.59 / 50.50	16.4	**90.7***	4.01	91.6 / 92.2

表2　WikiBio テストセットに対する評価．Fluency は *Mostly Fluent* を入れるか否かで 2 通りの数値を示した．
星印はブートストラップ・テストによりベースラインと比べ統計的有意（$p < .001$）を示す．

4　実験

WikiBio データセット [8] を使って以下を比較した：
BERT-to-BERT [12]：Transformer に基づくエンコーダー・デコーダーモデル [16] を BERT [4] チェックポイントで初期化したもの．
Structure-aware Seq2Seq [10]：インフォボックスのフィールド名とコンテンツを分けて扱う，LSTM に基づくモデル．WikiBio 評価で最高の BLEU を記録．
Pointer-Generator [13]：コピー機能付き Seq2Seq モデル（我々の実装）．
Confident BERT-to-RNN（提案モデル）：Transformer エンコーダーの BERT による初期化，および GRU [3] を使った自信度付きデコーダー．
Confident Pointer-Generator（提案モデル）コピー機能付き Seq2Seq に自信度付きデコーダー．

自動評価は BLEU と PARENT [5] を使い，人手評価はランダムに選んだ 1000 個の事例を各モデルの出力から同じく取り出し，5 人のアノテーターが *Faithfulness*（忠実性），*Coverage*，*Fluency*（流暢性）を採点した．

表2に結果を示す．人手評価によると提案手法は *Coverage* を少し低下させるも忠実性を大幅に改善した．提案した自信度の適切さを更に確かめるため，0.125 以下の自信度を持つトークンを削除する後処理を行うと，流暢性を少し損なうが忠実性の更なる上昇がみられた．自動評価指標については，PARENT Precision が忠実性と相関があると考えられ，提案手法は高い数値を記録した．一方 BLEU と忠実性の相関が見られなかった．

5　おわりに

入力データに忠実な文生成を促すため，我々は自信度付きデコーダーを提案し，WikiBio データセットにおいて生成文の忠実性を大幅に改善できることを示した．より詳細な実装，関連研究，人手評価の結果，アブレーションテストや生成文例などについては，[15] を参照されたい．

参考文献

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.
[2] M. Braverman, X. Chen, S. Kakade, K. Narasimhan, C. Zhang, and Y. Zhang. Calibration, entropy rates, and memory in language models. *CoRR*, abs/1906.05664, 2019.
[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.
[4] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.
[5] B. Dhingra, M. Faruqui, A. Parikh, M. Chang, D. Das, and W. Cohen. Handling divergent reference texts when evaluating table-to-text generation. In *ACL*, 2019.
[6] J. Gu, Z. Lu, H. Li, and V. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, Berlin, Germany, 2016.
[7] P. Koehn and R. Knowles. Six challenges for neural machine translation. In *WNMT*, 2017.
[8] R. Lebret, D. Grangier, and M. Auli. Neural text generation from structured data with application to the biography domain. In *EMNLP*, 2016.
[9] K. Lee, O. Firat, A. Agarwal, C. Fannjiang, and D. Sussillo. Hallucinations in neural machine translation. In *IRASL*, 2018.
[10] T. Liu, K. Wang, L. Sha, B. Chang, and Z. Sui. Table-to-text generation by structure-aware seq2seq learning. In *AAAI*, 2018.
[11] T. Luong, H. Pham, and C. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.
[12] S. Rothe, S. Narayan, and A. Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. *CoRR*, abs/1907.12461, 2019.
[13] A. See, P. Liu, and C. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL*, 2017.
[14] I. Sutskever, O. Vinyals, and Q. Le. Sequence to sequence learning with neural networks. In *NIPS*. 2014.
[15] R. Tian, S. Narayan, T. Sellam, and A. Parikh. Sticking to the facts: Confident decoding for faithful data-to-text generation, 2019.
[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*. 2017.
[17] O. Vinyals and Q. Le. A neural conversational model. *CoRR*, abs/1506.05869, 2015.
[18] S. Wiseman, S. Shieber, and A. Rush. Challenges in data-to-document generation. In *EMNLP*, 2017.