拡散モデルを用いたテキスト生成における「崩壊問題」と時刻埋め込みの影響

野坂 瞭太　松崎 拓也
東京理科大学大学院
1424519@ed.tus.ac.jp　matuzaki@rs.tus.ac.jp

概要

拡散モデルは，ランダムノイズに対して「ノイズ除去」を繰り返し行うことで徐々に良いサンプルを出力する，生成モデルの一種である．テキスト生成に応用する際の課題として，生成の過程において，ある回数以降のノイズ除去でサンプルの改善に失敗し，生成結果を崩壊させていく現象がある．本研究は，この現象に関して，これまで注目の薄かった時刻の埋め込みに焦点を当て，その影響を調査した．結果として，時刻の埋め込みがこの「崩壊問題」の一因であることを示し，崩壊を抑制する正則化を提案する．

1　はじめに

デノイジング拡散確率モデル（Denoising Diffusion Probabilistic Models; DDPMs）[1] は，画像をはじめとした連続データにおいて成功を収めた生成モデルで，埋め込み表現を介して離散データを生成する研究も盛んに行われている．DDPM は，まず時刻 T でランダムノイズをサンプリングし，以降時刻 0 に向けて「ノイズ除去」を段階的に行うことで，徐々に美しいサンプルを出力する．テキストを対象とする拡散言語モデル（Diffusion Language Models; DLMs）においては，一段階のノイズ除去を「ノイズを完全に除去し，小さいノイズを改めて付加する」ことで表現することが一般的である [2].

現在の DLM は品質に難があり実用レベルに至っていない．原因の一つとして，生成中，ある時刻を超えると「完全なノイズ除去」タスクに躓き始め，却ってサンプルを崩壊させていく現象が存在する [3]．本稿ではこれを **崩壊問題** と呼ぶ．「改めて付加するノイズ」は徐々に小さくするため，ノイズ除去タスクは生成の過程が進むのに伴って易しくなっていくべきであり，この振る舞いは期待に反している．先行研究では生成を複数回行い Minimum Bayes-Risk decoding [4] 等を用いて比較的良いサンプルを選定する試みが行われている [2] [5] [6] が，拡散モデルの長所である生成結果の多様性とのトレードオフがある．生成時にデータに付加するノイズの大きさをアドホックに調整することで崩壊を回避するような試み [3] は，DLM の可能性を示唆しつつも DDPM の枠組みから外れるため汎用性に乏しく，根本的な解決が待たれている．

生成の各ステップにおいてモデルにはノイズ付き単語埋め込みとともに現在時刻を表す時刻埋め込みを入力する．単語埋め込みに正則化を加えると一定の改善が見られるという報告 [3] はあるが，時刻埋め込みがモデルに与える影響は未だ明らかでない．

本研究は，拡散モデル本来の「徐々に美しいデータを生成する」挙動をテキスト生成において実現することを目的として時刻埋め込みに着目し，単語埋め込みとの関係を調査した．まず，時刻埋め込みが単語埋め込みを損壊する度合いを定義し，実際に情報を失わせていることを明らかにする．加えて，崩壊問題の解決策として両埋め込みに相互に作用する正則化を提案し，その有効性を示す．

2　準備

2.1　デノイジング拡散確率モデル

DDPM は以下の 2 種類の過程に基づく [1].

拡散過程　訓練データ $\mathbf{z}_0 \sim q(\mathbf{z}_0)$ に対し，次の Markov 過程で **ノイズ付加** を行う．

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\alpha_t}\mathbf{z}_{t-1}, \beta_t I\right)$$

$0 < \beta_1 < \cdots < \beta_T < 1$, $\alpha_t = 1 - \beta_t$ はノイズの大きさを決めるハイパーパラメータで，ノイズスケジュールという．$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, $\bar{\beta}_t = 1 - \bar{\alpha}_t$ とおくと

$$q(\mathbf{z}_t \mid \mathbf{z}_0) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, \bar{\beta}_t I\right) \tag{1}$$

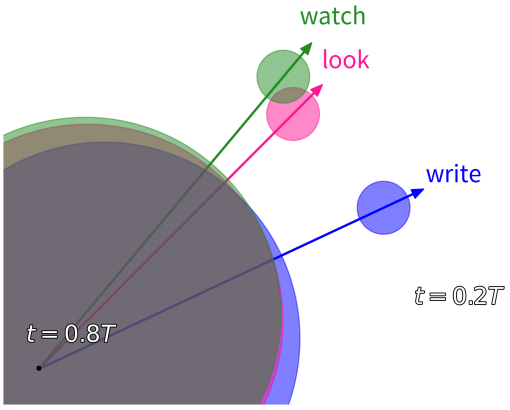

図 1　時刻 $0.2T$ および $0.8T$ におけるノイズ付加の様子．円はサンプリングされる主なベクトルの領域を表す（時刻が進むにつれ，原点方向に近づき，半径は大きくなる）．

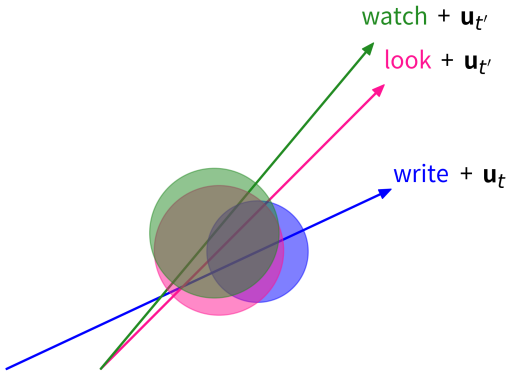

図 2　時刻埋め込みによる平行移動が分離を損なわせる様子．時刻 t においてノイズを付した “write” と時刻 t' においてノイズを付した “watch”, “look” が混同される．

が成り立つ．$\mathbf{z}_t$ が含む $\mathbf{z}_0$ の情報は次第に減少し，十分大きな T に対して $\mathbf{z}_T$ は $\mathcal{N}(0, I)$ に従うと見做せる．サンプリングした $\{\mathbf{z}_t\}_{t=0}^{T}$ を用いてノイズを除去するモデル $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$ を訓練する．

生成過程　ランダムノイズ $\mathbf{z}_T \sim \mathcal{N}(0, I)$ をサンプリングし，モデル $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$ を用いて **ノイズ除去** を時刻 $t = T, \ldots, 1$ において順に行うことで，新たなデータ $\mathbf{z}_0$ を出力する．

2.2　拡散言語モデル

DLM は，まず単語埋め込みの系列 $\mathbf{z}_0 = [\mathbf{z}_{0i}]_{i=1}^{L}$ を DDPM によって生成し，それを離散化することで単語列 $\mathbf{y} = [y_i]_{i=1}^{L}$ を得る [2]．生成過程において，時刻 t ではノイズを完全に除去した後 $t-1$ 回分のノイズ付加を行うことで一段階のノイズ除去を行う．

$$p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t) = q(\mathbf{z}_{t-1} \mid \mathbf{z}_0 = \mathbf{z}_\theta(\mathbf{z}_t, t))$$
$$\mathbf{z}_\theta(\mathbf{z}_t, t) = \mathrm{Transformer}(\mathbf{u}_\phi(\mathbf{z}_t, t))$$

$\mathbf{z}_\theta$ は任意の時刻 t のノイズ付き単語埋め込み列 $\mathbf{z}_t$ から $\mathbf{z}_0$ を予測するモデルで，Transformer [7] を利用する．$\mathbf{u}_\phi$ は $\mathbf{z}_t$ に時刻の情報を付加するモデルで，典型的には各 $\mathbf{z}_{ti}$ に時刻埋め込み $\mathbf{u}_t$ を足し合わせる．

$$\mathbf{u}_\phi(\mathbf{z}_t, t) = [\mathbf{z}_{ti} + \mathbf{u}_t]_{i=1}^{L}$$

ノイズ除去に関する損失 $\mathcal{L}_{\text{denoise}}$ は DDPM から導出される．単語列の生成モデルとしての DLM において，DDPM で生成する $\mathbf{z}_0$ は最終的な生成データである単語列 $\mathbf{y}$ に対する隠れ変数の役割を果たす．$\mathcal{L}_{\text{denoise}}$ は主に $\mathbf{z}_0$ と $\mathbf{z}_\theta(\mathbf{z}_t, t)$ の 2 乗誤差からなるので，$\mathcal{L}_{\text{denoise}}$ のみを小さくする方向に単語埋め込みを調整したとすると，全ての単語埋め込みが 1 点に集まるような無意味な配置が好まれると考えられる．

したがって，隠れ変数 $\mathbf{z}_{0i}$ と単語 y_i の適切な対応付けを学習するための信号として，ノイズ除去損失とは別に単語埋め込みが適切に分離されていることを促進するような損失を考える必要がある．そこで，単語埋め込み $\mathbf{w}_k$ を単語 $k \in \{1, \ldots, V\}$ に丸める確率分布を，すべての単語埋め込みとの内積の Softmax

$$p_\phi(k \mid \mathbf{w}_k) = \frac{\exp\langle \mathbf{w}_k, \mathbf{w}_k \rangle}{\sum_{\ell=1}^{V} \exp\langle \mathbf{w}_k, \mathbf{w}_\ell \rangle}$$

で定義し，訓練は $\mathcal{L}_{\text{denoise}}$ と p_ϕ の負の対数尤度 $\mathcal{L}_{\text{round}}$ との和 $\mathcal{L}_{\text{total}}$ を最小化する：

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{denoise}} + \mathcal{L}_{\text{round}}$$
$$\mathcal{L}_{\text{round}} = \mathbb{E}_{\mathbf{y}, \mathbf{z}_0}\left[-\sum_{i=1}^{L} \log p_\phi\left(y_i \mid \mathbf{z}_{0i}\right) \right].$$

3　手法

モデルは丸め損失によって隠れ変数 $\mathbf{z}_{0i}$ と単語 y_i の対応付けを学習するが，特にノイズが小さい時刻において，ノイズ除去の直接の対象となる $\mathbf{u}_\phi(\mathbf{z}_t, t)$ の各要素（以降 **入力ベクトル** と呼ぶ）が表す単語も正しく分離できなければならない．本節では，ノイズ除去のヒントとなるべき時刻埋め込みが実際にはノイズ付き単語埋め込みをさらに破壊してしまうことで Transformer がノイズ除去に失敗するという仮説を立て，検証方法および崩壊を抑制するための新しい正則化を説明する．

3.1　曖昧性スコア

ノイズ付加は単語埋め込みを原点方向に縮めガウシアンノイズを足すことで行われる（図 1）．モデルはその逆操作を学習するので，生成過程が進むにつれて入力 $\mathbf{z}_{ti}$ に対して出力される $\mathbf{z}_{0i}$ の分散が小さく

なっていき，次第に生成結果が確定する．しかし，Transformer に入力する際に時刻埋め込みによる平行移動が行われるため，実際の入力ベクトル $\mathbf{z}_{ti}+\mathbf{u}_t$ が異なる時刻における無関係な単語と混同される可能性がある（図 2）．ノイズ除去にあたって Transformer はすべての入力ベクトル $\mathbf{z}_{t1}+\mathbf{u}_t, \ldots, \mathbf{z}_{tL}+\mathbf{u}_t$ から時刻 t を暗黙的に推定していると考えられることから，本研究では，時刻 t における混同を $\mathbf{u}_t$ を中心として調査する．混同の度合いを 式 (2) で定義し，以降これを **曖昧性スコア** と呼ぶ．

$$\mathrm{Ambig}\,(t,t') = \frac{1}{V^2}\sum_{k=1}^{V}\sum_{\ell=1}^{V}\Big|\cos(\mathbf{w}_k,\mathbf{w}_\ell) - \cos\left(\sqrt{\bar{\alpha}_t}\mathbf{w}_k, \sqrt{\bar{\alpha}_{t'}}\mathbf{w}_\ell + \mathbf{u}_{t'} - \mathbf{u}_t\right)\Big| \quad (2)$$

cos は単語埋め込みの cos 類似度である．式 (1) より $\sqrt{\bar{\alpha}_t}\mathbf{w}_k$ は $q(\mathbf{z}_{ti} \mid \mathbf{z}_{0i} = \mathbf{w}_k)$ の期待値であるから，$\sqrt{\bar{\alpha}_t}\mathbf{w}_k + \mathbf{u}_t$ は時刻 t において単語 k を表す入力ベクトルの期待値にあたる．第 2 項の cos を，$\mathbf{u}_t$ を中心とした $\sqrt{\bar{\alpha}_t}\mathbf{w}_k + \mathbf{u}_t$ と $\sqrt{\bar{\alpha}_{t'}}\mathbf{w}_\ell + \mathbf{u}_{t'}$ の類似度と解釈すると，$\mathrm{Ambig}\,(t,t')$ は時刻 t から見たときに時刻 t' の各単語が表す言語的な意味が本来の意味から平均的にどの程度変化するかを表す．単語埋め込み空間と時刻埋め込み空間が直交しているとき $\mathrm{Ambig}\,(t,t') = 0$ となる．

3.2 分離を強調する正則化

入力ベクトルの分離を強く促進する正則化を $p_\theta(k \mid \mathbf{w}_k)$ の尤度に倣い 式 (3) で定義する．

$$\ell_{\mathrm{disambig}}(k,\mathbf{w}_k,t,t') = \frac{\exp\left\langle\sqrt{\bar{\alpha}_t}\mathbf{w}_k + \mathbf{u}_t, \sqrt{\bar{\alpha}_{t'}}\mathbf{w}_k + \mathbf{u}_{t'}\right\rangle}{\sum_{\ell=1}^{V}\exp\left\langle\sqrt{\bar{\alpha}_t}\mathbf{w}_k + \mathbf{u}_t, \sqrt{\bar{\alpha}_{t'}}\mathbf{w}_\ell + \mathbf{u}_{t'}\right\rangle} \quad (3)$$

式 (1) より $\sqrt{\bar{\alpha}_t}\mathbf{w}_k$ は $q(\mathbf{z}_{ti} \mid \mathbf{z}_{0i} = \mathbf{w}_k)$ の期待値であるから，ℓ_{disambig} を大きくすることは時刻 t において単語 k を表す入力ベクトルが時刻 t' における入力ベクトルと平均的に分離されることを促す．$\bar{\alpha}_0 = 1, \mathbf{u}_0 = \mathbf{0}$ と定めると $t = t' = 0$ のとき $p_\theta(k \mid \mathbf{w}_k)$ の尤度に一致するので，丸めに関する従来の損失の一般化になっている．本稿では，ノイズが多い時刻では混同をしてもよく，生成が進んだ時刻での混同を特に抑えたいという直感をもとに，$t' = 0$ に固定して実験を行う．したがって

$$\mathscr{L}_{\mathrm{disambig}} = \mathbb{E}_{\mathbf{y},\mathbf{z}_0,t}\left[-\sum_{i=1}^{L}\log \ell_{\mathrm{disambig}}(y_i,\mathbf{z}_{0i},t,0)\right]$$

を最小化する．

4 実験

4.1 実験設定

ノイズ除去モデル $\mathbf{z}_\theta$ として 12 層 Encoder-Decoder 型 Transformer を用いた．Encoder には言い換えあるいは平易化の元文を入力する．一般的な設定に従い，最大時刻 $T = 2000$，ノイズスケジュールは *sqrt* [2] とした（詳細は 付録 A）．

データセット　言い換え（Paraphrasing）タスクに Quora Question Pairs [8] を，平易化（Text Simplification）タスクに Wiki-Auto [9] を利用する．

評価指標　BLEU [10] および BERTScore [11] でタスクに対する精度を，Self-BLEU [12] で生成結果の多様性を評価する．

ベースライン　参考として標準的な DLM である DiffuSeq [5] と比較する．著者らが公開している生成例を評価した．本研究における $\mathscr{L}_{\mathrm{disambig}}$ を導入しなかった場合のモデルは DiffuSeq と類似のアーキテクチャであり同程度の性能が見込まれる．

4.2 結果と考察

最終的な生成結果の評価を 表 1 に，生成過程における途中経過の評価，すなわち各 $t = T, \ldots, 1$ に対する $\mathbf{z}_\theta(\mathbf{z}_t, t)$ の評価を 図 3 に示す．標準的な DLM では，生成品質を表す BLEU, BERTScore が悪化していく崩壊問題が起きていること（実例は 付録 B），提案手法によってそれを抑制できていることが確認できる．特に Quora Question Pairs では崩壊を抑制する以上にサンプルの品質を向上させる様子が見られる．

曖昧性スコアを 図 4 に示す．$t = t'$ に近づくほど曖昧性スコアは小さくなるが，これは近い時刻の埋め込み同士は近くなるよう自然に学習されるためと考えられる．いずれの場合も $t = t'$ から離れるほど混同が強まるが，$\mathscr{L}_{\mathrm{disambig}}$ の導入によって特にノイズが小さい時刻ほど曖昧性を解消できていることが認められる．

Self-BLEU は提案手法によって悪化し，一般に拡散モデルに期待される生成結果の多様性を損なう結果となった．しかし，図 3 によれば Self-BLEU の向上は $\mathscr{L}_{\mathrm{disambig}}$ の有無に関わらず崩壊（BLEU, BERTScore の悪化）に伴っており，これまで DLM に報告されてきた多様さはサンプルを破損させることによる見かけのものであることが示唆される．崩

表 1　最終ステップの出力の評価．括弧内は生成中の最良の値との差．

	Quora Question Pairs			Wiki-Auto		
	BLEU	BERTScore	Self-BLEU	BLEU	BERTScore	Self-BLEU
DiffuSeq	22.95	79.30	32.90	38.46	77.81	56.79
正則化なし	23.61 (−1.77)	79.59 (−1.90)	48.20 (+0.00)	36.67 (−7.24)	75.83 (−5.79)	63.18 (+0.00)
正則化あり	26.45 (−0.47)	82.44 (−1.03)	69.31 (+9.19)	40.49 (−1.57)	79.87 (−1.28)	87.75 (+9.87)

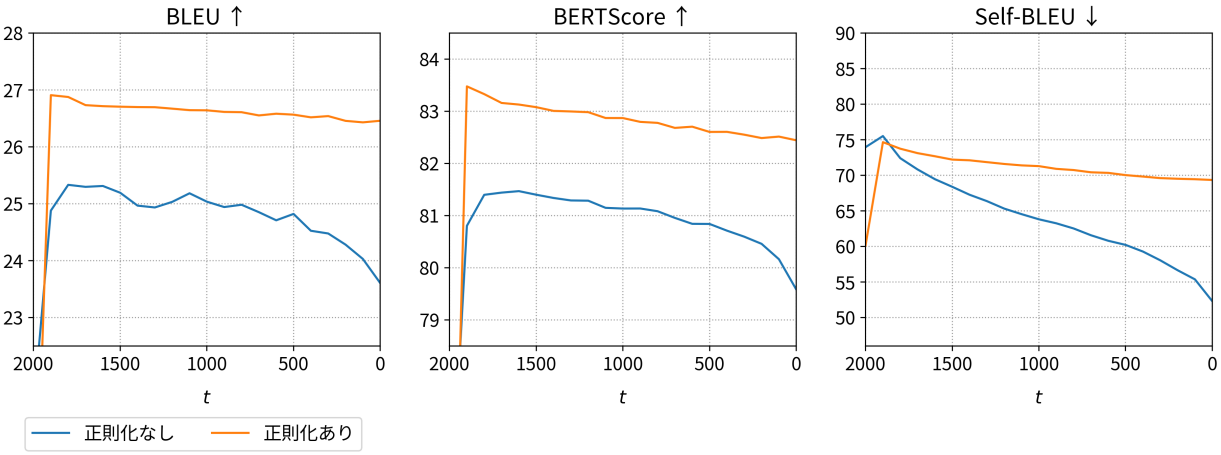

図 3　生成中の $\mathbf{z}_\theta(\mathbf{z}_t, t)$ の評価（Quora Question Pairs）

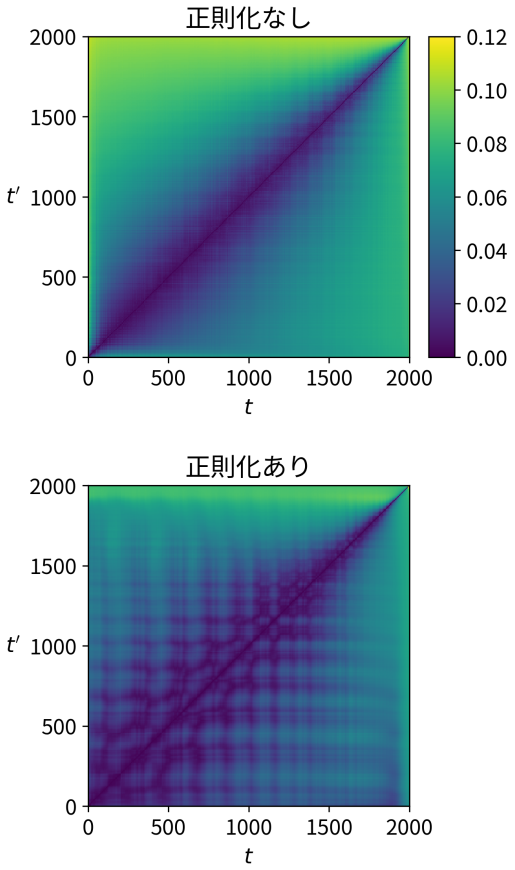

図 4　曖昧性スコア（Quora Question Pairs）

壊を許容したとしても，Self-BLEU は BLEU と同程度の値になることが理想的と考えられ，テキストを多様に生成するには拡散モデルにおいても一層の工夫が必要である．

BLEU, BERTScore に関しては生成過程の早い段階で最良のサンプルが現れ，半分以上の時刻が崩壊に費やされている．生成速度の向上を目的として時刻をスキップする試み [1] が広く行われているが，左記の事実は，代わりに生成過程を中断することが有効に働く可能性を示している [2]．ただし，最良のサンプルをランタイムに特定することは容易ではなく，崩壊問題の解決は依然として喫緊の課題である．

5　おわりに

本研究は，DLM の課題の一つである崩壊問題の解決を目指し，時刻埋め込みがその一因であることを指摘した．さらに，単語・時刻埋め込みの双方に及ぶ包括的な正則化を提案し，改善を確認した．今後の展望としては，さらに効果的な損失の設計や，平行移動以外の方法による時刻埋め込みの使用が挙げられる．

1)　ノイズ除去を $p_\theta(\mathbf{z}_{t-c} \mid \mathbf{z}_t) = q(\mathbf{z}_{t-c} \mid \mathbf{z}_0 = \mathbf{z}_\theta(\mathbf{z}_t, t))$ で定め，T/c ステップかけて生成する．

2)　類似の結果として，等間隔ではなくノイズが大きい時刻を多めに経由するようなスキップを行うと通常の生成過程よりも高品質なサンプルが得られるという報告 [13] がある．

参考文献

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In **Proceedings of the 34th International Conference on Neural Information Processing Systems**, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[2] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM Improves Controllable Text Generation. In **Proceedings of the 36th International Conference on Neural Information Processing Systems**, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

[3] Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. Empowering Diffusion Models on the Embedding Space for Text Generation. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 4664–4683, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[4] Shankar Kumar and William Byrne. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In **Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004**, pp. 169–176, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[5] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. In **The Eleventh International Conference on Learning Representations**, 2023.

[6] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. Text Diffusion Model with Encoder-Decoder Transformers for Sequence-to-Sequence Generation. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 22–39, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In **Proceedings of the 31st International Conference on Neural Information Processing Systems**, NIPS'17, p. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[8] DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. Quora Question Pairs, 2017. Kaggle.

[9] Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. Neural CRF Model for Sentence Alignment in Text Simplification. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7943–7960, Online, July 2020. Association for Computational Linguistics.

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[11] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In **International Conference on Learning Representations**, 2020.

[12] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A Benchmarking Platform for Text Generation Models. In **The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval**, SIGIR '18, pp. 1097–1100, New York, NY, USA, 2018. Association for Computing Machinery.

[13] Zecheng Tang, Pinzheng Wang, Keyan Zhou, Juntao Li, Ziqiang Cao, and Min Zhang. Can Diffusion Model Achieve Better Performance in Text Generation ? Bridging the Gap between Training and Inference ! In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Findings of the Association for Computational Linguistics: ACL 2023**, pp. 11359–11386, Toronto, Canada, July 2023. Association for Computational Linguistics.

表 2　ハイパーパラメータ

単語・時刻埋め込みの次元	128
Transformer の入出力の次元	768
Transformer のフィードフォワード層の次元	3072
入力の長さ	64
出力の長さ	64

表 3　崩壊問題が生じた生成過程の $\mathbf{z}_\theta(\mathbf{z}_t, t)$ と最終的な出力（Wiki-Auto）．特殊トークンは省略．

入力	Unicode is a computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems.
正解	Unicode is a standard for encoding computer text in most of the internationally used writing systems into bytes.
$t = 2000$	Unicode is a the in the for the and....
1800	Unicode is a computing industry standard for the consistent encoding,..
1600	Unicode is a computing industry standard for world's writing systems.
1400	Unicode is a computing industry of the world's writing systems.
1200	Unicode is a computing industry standard for the's writing systems.
1000	Unicode is one of the for some of the writing systems.
800	Unicode is one of the for in of the writing century.
600	Unicode is one at school for the of the is.
400	Unicode is one at school for local in the consistent Roman.
200	Unicode is one at school for the in the consistent the.
出力	Unicode is one at school for rain in one a Roman.
入力	The North London line (NLL) is a railway line which passes through the inner suburbs of west, north-west and north London, England between Richmond in the south-west and Stratford in the east, avoiding central London.
正解	The North London line is a railway line of the London Overground.
$t = 2000$	The North London line (NLL) is a the,.....
1800	The North London line is aLL, is a railway line.
1600	The North London line (NLL) is a railway line.
1400	The North London line (NLL) is a railway line.
1200	The North London line is anLL, north- railway line.
1000	The North London line (NLL) is a railway line.
800	The North London line is an a railway line in railway line.
600	The North London line is an important railway used in railway line.
400	The North London line is an, railway London in railway line.
200	The North London line is an R central London in Prime simple.
出力	The North London line is an of central London in theinson.

A　実験詳細

ハイパーパラメータは表 2 の通りである．単語埋め込み，時刻埋め込みはともに学習する．トークナイザには `facebook/bart-base` [3] を利用した．`<pad>` の埋め込みも学習し，出力の長さは `<pad>` の生成によって調整される．

BERTScore は `microsoft/deberta-xlarge-mnli` を用いて計算した．Self-BLEU の計算の際には生成を 3 回行った．曖昧性スコアはデータセットに含まれる単語の 1/10 から算出した．

B　崩壊の例

崩壊問題が生じた生成過程の実例を表 3 に示す．

3）　ここで挙げるモデルはすべて Hugging Face https://huggingface.co/ で公開されているものを指す．