訓練・推論時の不一致を解消する離散拡散テキスト生成モデル

浅田 真生 [1]　三輪 誠 [2,1]
[1] 産業技術総合研究所　[2] 豊田工業大学
masaki.asada@aist.go.jp
makoto-miwa@toyota-ti.ac.jp

概要

本研究では，離散拡散モデルを用いたテキスト生成において生じる訓練時と推論時の不一致に注目し，この問題を解消するため，訓練時の損失計算においてモデルが予測した系列を次ステップの入力として使用する 2 ステップ損失計算アプローチと，その確率的スケジューリングを提案する．提案手法を広く使用されている 4 つのテキスト生成ベンチマークデータセットにおいて学習・評価した結果，2 ステップ損失計算による離散拡散モデルの性能向上を確認した．

1　はじめに

近年，拡散モデルはテキスト生成モデルの分野において大きな注目を集めている [1, 2]．拡散モデルは拡散ステップ数を調整することで，テキスト生成の品質と速度のバランスを実現できることが知られており，系列全体を一度に生成する非自己回帰モデルよりも高い生成品質を示し，トークンごとに逐次生成を行う自己回帰モデルよりも高速な生成を可能にする [3]．テキスト生成における拡散モデルは大きく分けて連続拡散モデル [1, 2] と離散拡散モデル [4, 3] に分類される．特に離散拡散モデルは，ノイズ付与プロセスを，対象トークンをマスクトークンに置換するものとして解釈する．離散拡散モデルは，マスク補完を事前学習タスクとして行う事前学習言語モデルの性能を活用できる利点があり，いくつかのデータセットで既存の連続拡散モデルを上回る性能を示している [3]．

しかし，既存の離散拡散テキスト生成モデルには，訓練と推論の間に生じる不一致という問題が存在する．拡散モデルの訓練時には，ランダムに選ばれた特定の拡散ステップまで正解系列にノイズを加え，ノイズが付与された系列から正解系列を予測できるように損失を最小化する．一方で推論時には，

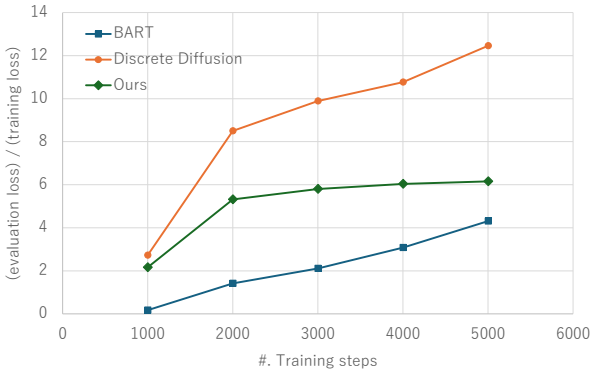

図 1　訓練ステップごとの（推論時の損失）/（訓練時の損失）の値．既存の離散拡散モデルでは自己回帰モデル (BART) と比較して推論時と訓練時の損失のギャップが大きいが，提案手法では軽減している．

前のステップで予測された系列にノイズを加える．訓練時は常にモデルの入力が正解系列にノイズ付与したものであるのに対し，推論時は常にモデルの入力は予測系列にノイズ付与したものであるという不一致があり，図 1 に示すように，既存の離散拡散モデルは訓練時に対する推論時の損失比が大きい．自己回帰型テキスト生成においては訓練時と推論時の不一致に対処する研究 [5] が行われてきたが，拡散テキスト生成においてこの問題に取り組むアプローチは未だ検討されていない．

本研究では，訓練と推論における不一致を解消するため，予測された系列を次のステップの入力として利用する 2 ステップの拡散学習手法を提案し，それに伴って，予測系列の利用による訓練初期段階の学習難度の過度な上昇を防ぐための確率的スケジューリング手法を導入する．

本研究の貢献は以下の通りである．[1)]

- 学習と推論の不一致に対処する新しい離散型拡散テキスト生成手法を提案
- 4 つのテキスト生成ベンチマークデータセット

1)　本論文は，COLING2025 にて発表する研究成果 [6] に基づくものである．

での実験結果を通じて，提案手法が離散拡散テキスト生成の性能向上に寄与することを確認

2 関連研究

拡散モデルによるテキスト生成は，大きく連続拡散モデルと離散拡散モデルの二つに分類される．連続拡散モデルはトークン埋め込みの潜在空間を利用し，ランダムなガウスノイズを徐々にターゲットトークンの埋め込み表現へと近づけ，その後埋め込み表現をトークン列に変換することでテキスト生成を行う [7]．一方で離散拡散モデルは，マスクトークンをノイズと解釈し，マスク穴埋めを複数ステップにわたって徐々に進めることでテキスト生成を実施する [8]．離散拡散モデルは，BART [9] といった事前学習済みテキスト生成モデルを活用できる利点を持つにもかかわらず研究が十分に進んでおらず，この分野はさらなる研究が必要である．そのため，本研究では離散拡散モデルによるテキスト生成の品質向上を目指す．本節では以降，Diffusion-NAT [3] で採用されているベースライン離散拡散テキスト生成モデルについて述べる．

離散拡散モデルによるテキスト生成は，条件付き確率 $P(Y|X)$ のモデル化として定式化される．ここで，$X = \{x_1, x_2, \cdots, x_m\}$ および $Y = \{y_1, y_2, \cdots, y_n\}$ は，それぞれ入力系列および出力系列を示し，トークン x および y は，語彙 $\mathcal{V}$ に属する．離散拡散モデルは，$K = |\mathcal{V}|$ 個のカテゴリを持つ離散確率変数を用いて，ノイズ付与プロセスを以下のように実行する：

$$q(Y_t|Y_{t-1}) = v^\top(Y_t)\mathbf{Q}_t v(Y_{t-1}). \tag{1}$$

ここで，$v(Y)$ は各トークンを K 次元の one-hot ベクトルに変換する写像ベクトル，$\mathbf{Q}_t$ は遷移確率行列を表し，$[\mathbf{Q}_t]_{i,j}$ はトークン i がトークン j に置き換えられる確率を示す．具体的には，ステップ t にてトークン i が [MASK] トークンでない場合，α_t の確率で変更されず，γ_t の確率で [MASK] トークンに置き換えられ，残りの確率 $\beta_t = 1 - \alpha_t - \gamma_t$ で語彙 $\mathcal{V}$ 中の他のトークンに遷移する：

$$[\mathbf{Q}_t]_{i,j} = \begin{cases} \alpha_t & \text{if } j = i, \\ \gamma_t & \text{if } j = \texttt{[MASK]}, \\ \beta_t & \text{otherwise}, \end{cases} \tag{2}$$

ここで，α_t および γ_t は事前に定義されたノイズスケジューラによって決定される．

離散拡散モデルでは，事前学習言語モデルを用いて，各拡散ステップでノイズが加えられたトークンを復元する．この際，事前学習モデルを非自己回帰的にテキスト生成を行うよう修正し，すべてのマスクされたトークンを同時に復元できるようにする．具体的には，t ステップにおいて，ソース系列 X と [MASK] トークンを含むノイズ付加済みターゲット系列 Y_t をそれぞれ事前学習モデルのエンコーダとデコーダに入力し，全ての [MASK] トークンをノイズ除去する．

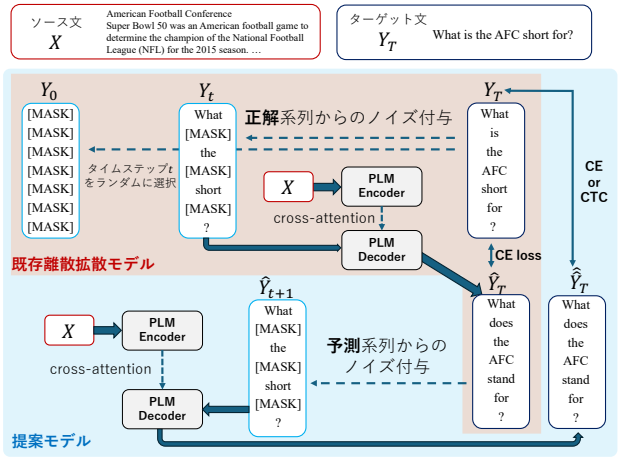

図 2 Training workflow of the discrete diffusion model with step-aware loss compared to the existing model

訓練中，モデルは各拡散ステップで事前学習モデルの非自己回帰型生成により元の全トークン $Y_0 = \{y_1^{(0)}, \cdots, y_n^{(0)}\}$ を予測する：

$$\text{PLM}(\{y_1^{(t)}, \cdots, \texttt{[MASK]}\}, X) = \{\hat{y}_1^{(0)}, \cdots, \hat{y}_n^{(0)}\}, \tag{3}$$

拡散ステップ t は各サンプルに対してランダムに一度だけ選ばれ，X と $Y_t = \{y_1^{(t)}, \cdots, \texttt{[MASK]}\}$ が，それぞれ事前学習モデルのエンコーダとデコーダに入力される．入力 X の表現埋め込みはクロスアテンションを通じてデコーダに渡される．損失関数として交差エントロピー損失が用いられ，以下のように表される：

$$L_Y = -\sum_{i=1}^{n} \log p_\theta(y_i^{(0)}|Y_t, X). \tag{4}$$

推論時には，与えられた Y_t に基づき $\hat{Y}_0$ を推定し，その後 $t-1$ ステップのノイズを追加して Y_{t-1} を生成する．このプロセスを最終的に Y_0 が得られるまで繰り返す．

3 提案手法

本研究では，離散拡散モデルの訓練と推論における不一致を解消するため，モデルの予測系列を次の

ステップの入力として利用する 2 ステップの損失計算手法とその確率的スケジューリング手法を提案する．提案モデルの概要を図 2 に示す．

2 ステップ損失　節 2 で述べたように，訓練中にタイムステップ t がランダムに選択され，Y_0 から t ステップまでノイズを付与した Y_t が準備され，そこから $\hat{Y}_0$ が予測される．つまり，訓練中は，モデルは常に**正解**系列にノイズ付与された系列を受け取る．一方，推論中は，モデルは Y_T（T は総拡散ステップ数）から始め，$\hat{Y}_0$ を予測し，次に $\mathbf{Q}_t$ を予測系列に適用して $T-1$ ステップ目の入力を準備し，この処理を拡散ステップが 0 になるまで繰り返す．つまり，モデルは常に**予測**系列からノイズ付与した系列を入力として受け取る．この学習と推論の不一致に対処するために，2 ステップの拡散プロセスを損失計算に利用する手法を提案する．訓練時にランダムに拡散ステップ t が選ばれ，デコーダへの入力 Y_t が $\mathbf{Q}_t$ を Y_0 に適用して準備され，その後モデルは以下のように $\hat{Y}_0$ を予測する:

$$\hat{Y}_0 = \mathrm{PLM}(Y_t, X). \tag{5}$$

提案モデルでは，$\hat{Y}_0$ を損失計算に用いるのではなく，$\hat{Y}_0$ にノイズを付与し，次の拡散ステップの入力として利用する．これにより，訓練時においてもモデルが自己予測した系列からノイズ付与した系列を入力として受け取れるようになる．具体的には，$\hat{Y}_{t-1}$ は $\mathbf{Q}_t$ を用いて準備され，その結果得られた系列 $\hat{Y}_0$ を使って，モデルは以下のようにターゲット系列を予測する:

$$\hat{\hat{Y}}_0 = \mathrm{PLM}(\hat{Y}_{t-1}, X), \tag{6}$$

ここで $\hat{\hat{Y}}_0 = \{\hat{\hat{y}}_1^{(0)}, \cdots, \hat{\hat{y}}_n^{(0)}\}$ となる．交差エントロピー損失関数は次のように表される:

$$L_{\mathrm{CE}} = -\sum_i^n y_i^{(0)} \log \hat{\hat{y}}_i^{(0)}. \tag{7}$$

また，本研究では非自己回帰型テキスト生成で成功を収めている CTC 損失 [10] の離散テキスト生成への適用可能性も調査する．CTC を用いた変種の損失関数は次のように表される:

$$L_{\mathrm{CTC}} = -\log P_{\mathrm{CTC}}(Y_0|\hat{\hat{Y}}_0). \tag{8}$$

予測系列利用の確率的スケジューリング　式 6 による訓練時の自己予測系列の利用は，学習の初期段階ではモデルは誤りの多い予測系列を次ステップの入力とするため，過度に難しいタスクとなる．そのため，次の拡散ステップにおけるモデルの入力は，正解系列または自己予測系列のどちらかを確率的に選択する手法を提案する．ここで自己予測系列が選択される確率 p_k は学習が進むにつれて大きくなるように設定する．

$$\hat{\hat{Y}}_0 = \begin{cases} \mathrm{PLM}(\hat{Y}_{t-1}, X) & \text{with } p_k \\ \mathrm{PLM}(Y_t, X) & \text{with } 1-p_k, \end{cases} \tag{9}$$

ここで，$\hat{Y}_{t-1}$ は予測系列からのノイズ付き系列であり，Y_t は正解系列からのノイズ付き系列である．選択確率 p_k は，現在の学習ステップ k と総学習ステップ数 K に基づいて線形に決定される：$p_k = \frac{k}{K}$.

	XSum			MSNews		
	R-1	R-2	R-L	R-1	R-2	R-L
自己回帰型						
LSTM	-	-	-	30.0	14.6	27.7
Transformer	30.6	10.8	24.4	33.0	15.4	30.0
MASS	39.7	17.2	31.9	-	-	-
ProphetNet	39.8	17.1	32.0	-	-	-
BART	38.7	16.1	30.6	41.8	23.1	38.3
非自己回帰型						
NAT	24.0	3.88	20.3	-	-	-
iNAT	24.0	3.99	20.3	-	-	-
CMLM	23.8	3.60	20.1	-	-	-
LevT	24.7	4.18	20.8	-	-	-
BANG	32.5	8.98	27.4	32.7	16.1	30.3
ELMER	38.3	14.1	29.9	35.6	16.1	32.5
BnB	36.1	13.4	30.0	-	-	-
拡散モデル						
GENIE	29.3	8.3	21.9	-	-	-
AR-Diff	32.2	10.6	25.2	-	-	-
Diff-NAT	**38.8**	**15.3**	30.8	46.8	31.6	44.2
提案手法	38.5	14.8	**30.9**	**50.5**	**35.1**	**48.0**

表 1　要約タスクにおける性能比較．R-L, B-4, MT はそれぞれ ROUGE-L, BLEU-4, METOR を示す．太字は非自己回帰モデルおよび拡散モデルの中で最も高いスコアを示す．

4　実験

4.1　実験設定

2 つのテキスト要約タスクデータセット：XSum, MSNews および 2 つの質問生成タスクデータセット：SQuAD v1.1, MSQG を用いて提案した離散拡散モデルのファインチューニングおよび評価を行なった．データセットおよびモデル設定の詳細情報を，付録 A および B に，比較する既存モデルの詳細を付録 C に示す．

	SQuAD v1.1			MSQG		
	R-L	B-4	MT	R-L	B-4	MT
自己回帰型						
LSTM	-	-	-	25.3	3.5	14.1
Transformer	29.4	4.61	9.86	29.3	5.1	16.6
MASS	49.4	20.1	24.4	-	-	-
ProphetNet	48.0	19.5	23.9	-	-	-
BART	42.5	17.0	23.1	38.1	10.2	22.1
非自己回帰型						
NAT	31.5	2.46	8.86	-	-	-
iNAT	32.4	2.33	8.84	-	-	-
CMLM	31.5	2.51	8.85	-	-	-
LevT	31.3	2.27	9.14	-	-	-
BANG	44.0	12.7	18.9	33.1	**11.0**	18.4
ELMER	40.2	13.4	20.0	26.6	5.00	15.7
BnB	41.7	13.8	-	-	-	-
拡散モデル						
Diff-NAT	**46.6**	**16.1**	21.9	33.3	6.6	19.3
提案手法	43.5	15.4	**23.0**	**39.0**	8.0	**20.5**

表 2　質問生成タスクにおける性能比較．R-L, B-4, MT はそれぞれ ROUGE-L, BLEU-4, METOR を示す．太字は非自己回帰モデルおよび拡散モデルの中で最も高いスコアを示す．

Models	MSNews			MSQG		
	R-1	R-2	R-L	R-L	B-4	MT
CTC	**50.55**	**35.15**	**48.04**	**39.00**	8.09	**20.56**
-w/o 2-step	48.69	33.71	45.98	36.82	6.62	19.86
-w/o p_k	40.17	28.14	39.34	26.26	1.58	10.02
CE	50.14	34.70	47.58	38.96	**8.49**	20.30
-w/o 2-step	50.12	34.69	47.55	38.82	8.31	20.29
-w/o p_k	49.35	34.06	47.04	30.16	3.14	13.61

表 3　提案モデルの要素検証．2-step, p_k は 2 ステップ損失，p_k スケジューリングをそれぞれ意味する．

4.2　結果

表 1 は，テキスト要約タスクデータセット XSum および MSNews における提案手法と既存モデルの性能を比較している．XSum データセットでは，提案手法は Diffusion-NAT に比肩する性能を示している．MSNews データセットでは，提案手法が最も高い性能を示しており，自己回帰型生成モデルよりも高い性能を示している．

表 2 は，質問生成タスクデータセット SQuAD および MSQG における提案手法と既存モデルの性能を比較している．SQuAD データセットでは，提案手法は Diffusion-NAT と比較して ROUGE-L および BLEU-4 で低い値を示したが，METEOR では高い値を示した．MSQG データセットでは，提案手法がすべての評価指標において Diffusion-NAT を上回っており，また ROUGE-L スコアにおいては，提案手法が自己回帰モデルを含むすべてのモデルの中で最も高い性能を示した．

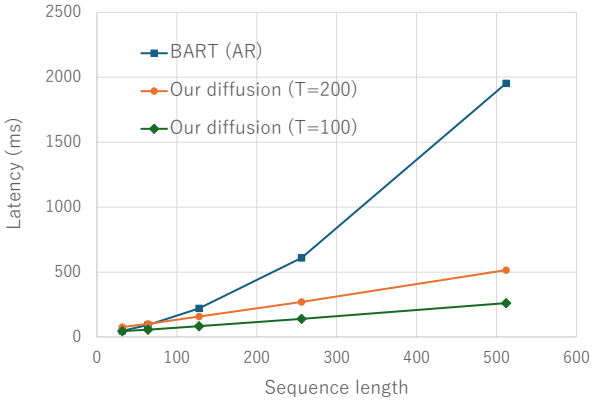

図 3　生成系列長に対する生成時間

表 3 は，2 ステップ損失とその確率的スケジューリングについて，CTC 損失および交差エントロピー損失の両方に対するアブレーション研究を示している．2 ステップ損失の導入がどちらの損失関数においても性能向上に貢献しており，また 2 ステップ損失利用における p_k スケジューリングの重要性を示している．特に CTC 損失を採用した場合は提案アプローチが大きく性能向上に寄与しており，結果として，CTC 損失と 2 ステップ損失計算および p_k スケジューリングを取り入れた手法が MSQG の BLEU-4 を除いて最も高い性能を示した．

4.3　系列長に対する生成速度

図 3 は，ターゲット系列長に対する自己回帰型モデル BART と提案モデルの推論速度を示している．この図より，BART は長い文を生成するにつれて劇的に速度が遅くなるのに対し，離散拡散モデルはより一貫した速度を維持することが分かる．これは，最大系列長が増加しても，各拡散ステップで行われるデコードが非自己回帰的に行われるためである．

5　おわりに

本研究では，離散拡散モデルにおける訓練と推論の間の不一致に対処することを目的として，2 ステップ損失と，その確率的スケジューリングを提案した．実験結果より，提案したアプローチがいずれも離散拡散モデルの性能向上に寄与することが示された．他の離散拡散モデルと比較した場合は提案手法が上回るまたは比肩する性能を示しており，いくつかの設定では自己回帰型生成モデルよりも高い性能を示した．今後の予定として，離散拡散モデルの大規模事前学習を実施したい．

謝辞

この成果は, 国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務（JPNP20006）の結果得られたものです.

参考文献

[1] Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In **ICML 2023**, pp. 21051–21064. PMLR, 2023.

[2] Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. **NeurIPS 2023**, Vol. 36, pp. 39957–39974, 2023.

[3] Kun Zhou, Yifan Li, Xin Zhao, and Ji-Rong Wen. Diffusion-NAT: Self-prompting discrete diffusion for non-autoregressive text generation. In Yvette Graham and Matthew Purver, editors, **EACL 2024**, pp. 1438–1451, St. Julian's, Malta, March 2024. ACL.

[4] Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. DiffusionBERT: Improving generative masked language models with diffusion models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **ACL 2023**, pp. 4521–4534, Toronto, Canada, July 2023. ACL.

[5] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **ACL 2019**, pp. 4334–4343, Florence, Italy, July 2019. ACL.

[6] Masaki Asada and Makoto Miwa. Addressing the training-inference discrepancy in discrete diffusion for text generation. In **COLING 2025**, Abu Dhabi, UAE, January 2025. ICCL.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. **NeurIPS 2020**, Vol. 33, pp. 6840–6851, 2020.

[8] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, **NeurIPS 2021**, Vol. 34, pp. 17981–17993. Curran Associates, Inc., 2021.

[9] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, **ACL 2020**, pp. 7871–7880, Online, July 2020. ACL.

[10] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In **ICML 2006**, pp. 369–376, 2006.

[11] Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. ELMER: A non-autoregressive pre-trained language model for efficient and effective text generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **EMNLP 2022**, pp. 1044–1058, Abu Dhabi, United Arab Emirates, December 2022. ACL.

[12] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In **ICLR 2021**, 2021.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **NeurIPS 2017**, p. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[14] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MASS: Masked sequence to sequence pre-training for language generation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, **ICML 2019**, Vol. 97, pp. 5926–5936. PMLR, 09–15 Jun 2019.

[15] Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. ProphetNet: Predicting future n-gram for sequence-to-Sequence Pre-training. In Trevor Cohn, Yulan He, and Yang Liu, editors, **Findings of EMNLP 2020**, pp. 2401–2410, Online, November 2020. ACL.

[16] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In **ICLR 2018**, 2018.

[17] Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, **EMNLP 2018**, pp. 1173–1182, Brussels, Belgium, October-November 2018. ACL.

[18] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, **EMNLP-IJCNLP 2019**, pp. 6112–6121, Hong Kong, China, November 2019. ACL.

[19] Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, **NeurIPS 2019**, Vol. 32. Curran Associates, Inc., 2019.

[20] Weizhen Qi, Yeyun Gong, Jian Jiao, Yu Yan, Weizhu Chen, Dayiheng Liu, Kewen Tang, Houqiang Li, Jiusheng Chen, Ruofei Zhang, Ming Zhou, and Nan Duan. Bang: Bridging autoregressive and non-autoregressive generation with large scale pretraining. In Marina Meila and Tong Zhang, editors, **ICML 2021**, Vol. 139, pp. 8630–8639. PMLR, 18–24 Jul 2021.

[21] Mohammad Golam Sohrab, Masaki Asada, Matīss Rikters, and Makoto Miwa. Bert-nar-bert: A non-autoregressive pre-trained sequence-to-sequence model leveraging bert checkpoints. **IEEE Access**, Vol. 12, pp. 23–33, 2024.

A　データセット

本節では，テキスト要約および質問生成タスクのタスク設定およびベンチマークデータセットについて説明する．

- テキスト要約タスクのベンチマークセットとして XSum データセットと MSNews データセットを使用した．評価指標として，ROUGE-1/2/L を用いた
- 質問生成は与えられたパラグラフと回答に基づいて質問文を生成するタスクである．ベンチマークセットとして SQUAD v1.1 と MSQG データセットを使用した．評価指標は，BLEU-4 , ROUGE-L，METEOR である

これらのデータセットの統計を，表 4 に示す．

データ分割設定およびこれらの評価指標の実装は，既存研究 [11, 3] に準拠しており，公平な比較のために，すべてのベースラインについて [3] で報告されたスコアを示している．

B　モデル設定

ノイズ付与を制御する $\mathbf{Q}_t$ の確率は線形スケジューラ [7] によって決定した．提案モデルは，Diffusion-NAT [3] に従い，BART-base のチェックポイントで初期化されており，139M のパラメータを持ち，追加のパラメータは含まれない．訓練中は，拡散ステップを 1,000 に設定した．推論時には，DDIM [12] を使用して推論拡散ステップを 200 にサンプリングした．

最適化には AdamW を使用し，学習率は 5e-5 とした．SQuAD v1.1 および MSQG データセットに対しては学習ステップ数を 10,000 に設定し，XSum および MSNews データセットに対してはそれぞれ 80,000 および 20,000 ステップを使用した．すべてのデータセットにおいて，グローバルバッチサイズを 512 に設定した．学習は NVIDIA V100 GPU 16 基を用いて実施した。

生成速度の比較において，ソース系列は長さ 512 にパディングし，BART の生成トークン数と提案拡散モデルのターゲット系列のパディング長を変化させることで速度比較を実施した．すべてのモデルでバッチサイズを 16 に設定し，実験は NVIDIA V100 GPU 1 基上で行った．

Task	Dataset	#.Train	#.Valid	#.Test
要約	XSUM	204,045	11,332	11,334
	MSNews	136,082	7,496	7,562
質問生成	SQUAD v1.1	75,722	10,570	11,877
	MSQG	198,058	11,008	11,022

表 4　要約データセットと質問生成データセットの統計

C　ベースラインの詳細

自己回帰型モデル

- Transformer [13]: 事前学習なしで自己回帰生成を行うモデル
- MASS [14]: エンコーダ・デコーダ構造からマスク系列復元事前学習を行うモデル
- BART [9]: 様々な系列変換タスクによって事前学習を行うモデル
- ProphetNet [15]: n-gram 予測を事前学習タスクとして実施するモデル

非自己回帰型モデル

- NAT [16]: 最初に提案された非自己回帰型テキスト生成モデル
- iNAT [17]: Iterative refinement を採用した非自己回帰モデル
- CMLM [18]: マスク穴埋めを繰り返す非自己回帰モデル
- Levenshtein Transformer (LevT) [19]: 編集に基づく非自己回帰モデル
- BANG [20]: 自己回帰型，非自己回帰型および半非自己回帰型生成をすべてサポートするモデル
- ELMER [11]: Transformer を用いた事前学習モデルパラメータにより非自己回帰型生成の性能を向上させたモデル
- BERT-nar-BERT (BnB) [21]: 既存のエンコーダのみモデルのチェックポイントを組み合わせて非自己回帰型生成を行うモデル

拡散モデル

- GENIE [1]: 事前学習した連続離散モデル
- AR-Diffusion (AR-Diff) [2]: 自己回帰デコードにより生成性能を向上した連続離散モデル
- Diffusion-NAT (Diff-NAT) [3]: 事前学習モデルのパラメータを活用した離散型拡散モデル．訓練と推論を複数イテレーション実施する自己プロンプティング手法を採用して生成テキストの品質は向上したが，生成速度は低下した