言語モデルが生成したテキストを書き換えるタスク非依存の復号手法

藤田 正悟 [1]　小林 尚輝 [1]　後藤 啓介 [1]
[1] 株式会社 LegalOn Technologies
{shogo.fujita,naoki.kobayashi,keisuke.goto}@legalontech.jp

概要

言語モデルは流暢なテキスト生成を可能とし幅広い生成タスクに使用されているが，その出力は常に正しいとは限らない．そのため，実際には言語モデルによって生成されたテキストを手作業で校正する必要があるが，これは時間のかかる作業である．本稿では指定された区間のテキストを書き換えるタスク，および書き換えのための復号手法の拡張を提案する．提案手法は書き換えたいテキストの生成に使用した言語モデルを再利用するため，追加のモデルの学習やデータセットを必要とせずテキストを書き換えることができる．実験では要約タスクと翻訳タスクにおける本手法の有効性を評価し，text infilling モデルと paraphrase モデルをベースラインとして比較した．提案手法は品質と多様性の両方においてベースラインを上回り，書き換えタスクへの有効性を実証した．

1　はじめに

近年，言語モデルのテキスト生成能力は飛躍的に改善され，その流暢な出力から機械翻訳や文書要約，チャットボットなどの幅広いテキスト生成タスクに利用されている．しかし，言語モデルによって生成されたテキストが常に正しいとは限らず，文法的な誤りや事実と異なる内容が含まれる可能性がある．このような問題から，言語モデルが生成したテキストを人手により校正する必要があるが，これはコストのかかる作業である．人手によるテキストの校正は，(1) 誤りを含む修正区間の特定，(2) 特定した区間のテキストの書き換えの二つのステップに分けられる．多くの場合，(1) の修正区間は直感的に判断でき，そのコストは比較的小さい．一方で (2) のテキストの書き換えは創造的なプロセスであるため，そのコストは比較的大きい [1]．したがって，(2) の書き換えが校正にかかるコストの大半を占める．

本稿では，校正にかかるコストの削減を目的として，言語モデルが生成したテキストを対象に，指定した区間を書き換えるタスクを提案する．ただし，本タスクで対象とする言語モデルは，要約や翻訳などの特定のタスクに応じたテキストの生成を目的とした言語モデルに限定し，書き換えられたテキストはそのタスクに応じた特性を満たす必要がある．例えば，文書要約を行う言語モデルの生成したテキストを対象にした書き換えでは，書き換えられたテキストも入力文書の要約である必要がある．そのため，一般的な知識から一部が欠落したテキストの穴埋めを行う text-infilling [2] や，テキストの意味を変えずに長さや文体の異なるテキストを生成する paraphrasing [3, 4] とはタスクの性質が異なる．また，実際のシステムとして運用する場合，複数の書き換え候補から手動で最も良いものを選べた方が有用である．そこで，本タスクでは複数の候補を生成し，その多様性も評価する．

先述したように提案する書き換えタスクは，特定の目的に沿った書き換えテキストを生成し，また言語モデルが生成したテキスト上の任意の区間の書き換えを扱う必要がある．これらの要件を満たすために，提案手法は元のテキストを生成した言語モデルを再利用し，一般的な復号アルゴリズムである Beam search を拡張して任意の書き換え区間において複数の書き換え候補を生成する．言語モデルの再利用により，元の目的に沿った書き換えテキストの生成を保証する．さらに言語モデルの再利用には，追加のデータセットやモデルを必要としない利点がある．また Beam search の拡張により書き換え区間の後方にあるテキストを生成する制約を与えることで，元の文脈と自然に繋がる書き換えテキストの生成を可能とする．

実験では，要約と翻訳の 2 つのタスクにおいて，

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

タスク固有の性能と書き換え候補の多様性の 2 つの観点から評価を行った．ベースライン手法として，text-infilling により書き換えを行う BART [5] と paraphrase により書き換えを行う DEltIteraTeR [4] を提案手法と比較し，提案手法がタスク固有の性能と多様性の両方においてベースラインよりも優れた性能を示した．

2 提案手法

2.1 タスク定義

提案する書き換えタスクを，言語モデルの生成したテキスト上の指定された区間に対して書き換え候補を生成するタスクとして定式化する．本タスクの入力は，特定のタスクで学習された言語モデル，言語モデルに入力されたテキスト，言語モデルの生成したテキスト $O = [o_1, \ldots, o_n]$ および O 上の書き換え区間 (s, e) とする．本タスクの出力は，$[o_s, \ldots, o_{e-1}]$ に代わる書き換え候補とする．以降では $[o_1, \ldots, o_{s-1}]$，$[o_s, \ldots, o_{e-1}]$，$[o_e, \ldots, o_n]$ をそれぞれ，prefix，target，suffix とし，prefix と suffix は書き換え区間の前後のテキスト，target は書き換え区間に該当するテキストである．

2.2 Suffix-Aware Generation (SAGen)

一般に，言語モデルは左から右へ逐次的にトークンを生成し，(eos) が出力されると生成を終了する．テキスト上の与えられた区間の書き換え候補を生成する場合，出力される書き換え候補は元のテキストの前後の文脈と自然につながる必要がある．しかし，先に述べたように，言語モデルは左から右へと生成するため，生成時に後ろの文脈を考慮しない．結果として生成された書き換え候補は，後続する文と上手く繋がらない．この問題を解決するために，我々は (eos) の代わりに suffix を出力して生成を終了する Suffix-Aware Generation (SAGen) を提案する．SAGen は，各時刻ステップにおいて，以下のように suffix を生成するスコアを計算する．

$$P(\text{suffix}|w_{<t}) = \prod_{i=e}^{n} P(o_i|w_1, \ldots, w_{t-1}, o_e, \ldots, o_{i-1})^{\frac{1}{|\text{suffix}|}} \quad (1)$$

ここで，P は言語モデルの生成スコア，$w_{<t}$ は時刻 t 以前に生成されたトークン列，$|\text{suffix}|$ は suffix のトークン数であり，スコアの正規化に用いられる．

2.3 出力の品質に関する工夫

SAGen は，(eos) の代わりに suffix を生成することで，書き換えられた出力の末尾を suffix にする．これは，出力を冗長にしたり，文法的に不正確にしたりする可能性があるため，これらの悪影響を回避するために以下の方法を提案する．

2.3.1 Length Adjust (LA)

書き換えタスクの目的は，元のテキストを置き換えるテキストを生成することであり，ほとんどの場合，書き換えの前後でテキストの長さは類似している．Length Adjust (LA) は，式 (1) で生成された suffix のスコアに α を用いて以下のように重み付けを行い，出力長を制御する．

$$\hat{P}(\text{suffix}|w_{<t}) = P(\text{suffix}|w_{<t})^{\alpha} \quad (2)$$
$$\alpha = \alpha_s + (\alpha_e - \alpha_s)\frac{\min(t-s, |\text{target}|)}{|\text{target}|}$$

α_s と α_e はそれぞれ，α の最大値と最小値を表すハイパーパラメータである．

2.3.2 Word Joint (WJ)

SAGen は suffix と文法的に繋がらないような文を生成してしまうことがある．これは，suffix の先頭のトークンの生成スコア $P(o_e|w_{<t})$ と，式 (1) との間に乖離があるために起こると考えられる．このずれを埋めるために，Word Joint (WJ) は，suffix の先頭の生成スコアを利用して suffix の生成確率を次のように補正する．

$$\hat{P}(\text{suffix}|w_{<t}) = \begin{cases} P(\text{suffix}|w_{<t}) & P(o_e|w_{<t}) \geq d \\ 0 & P(o_e|w_{<t}) < d \end{cases} \quad (3)$$

ここで，d は suffix に繋がるかどうかを判断する閾値でありハイパーパラメータである．

2.4 出力の多様性に関する工夫

実際の書き換えシステムのユースケースでは，複数の候補を出力させ，その中から最適なものを手動で選択する方法が考えられる．そこで，Beam search（BS）に SAGen を適用することで，suffix で終わる複数の候補を効率的に生成することができる．しかし，BS は尤度に基づいて候補を生成するため，ビーム間で類似した候補が生成される傾向がある．Diverse beam search (DBS) [6] は，ビームを複数のグ

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

ループに分割し，グループ間の類似性を低減する制約により，BS の出力間の多様性を改善する手法である．ここで DBS に追加して候補間の長さの多様性を改善する多様長ビーム探索（DLBS）を提案する．様々な長さの候補を生成することは，書き換え時の情報の追加や削除に相当するため有用である．DLBS はグループ g について式 (2) の α を計算する．

$$\alpha_g = \alpha_s + (\alpha_e - \alpha_s)\frac{\min(t-s, \beta_g|\text{target}|)}{\beta_g|\text{target}|} \quad (4)$$

ここで，β_g はグループ g の出力長を制御するパラメータである．

3 実験設定

準備: 我々は翻訳と要約における書き換えタスクに取り組む．まず，各タスクの言語モデルによる出力を用意する．書き換え区間の指定を模倣するため，文頭から 20%の単語を prefix，文末の 20%を suffix とした．残りの 60%を書き換え，各タスクに対する書き換えの適切性を評価した．

検証するデータセットと言語モデル: 翻訳タスクの検証では，データセットに WMT19 En-De [7]，言語モデルに ‘facebook/bart-wmt19-de-en’ を用いて実験した[1)]．また，要約タスクの検証では，データセットに XSum [8] を，言語モデルに ‘facebook/bart-large-xsum’ を用いた．実験では，各データセットに含まれる dev セットと test セットの分割をそのまま使用した．

評価指標: 各タスクの性能を評価する評価指標として，要約には ROUGE 1/2/L [9] を，翻訳には BLUE [10] と Comet22 [11] を使用した[2)]．これらのスコアは，各タスクの正解文と書き換えた出力の間で計算する．また，ChatGPT[3)] を利用した評価指標である ChatGPT-Score も報告する[4)]．書き換え候補の多様性を評価するために，生成された候補間の類似度を計算する Self-BLEU [14] を用いた．Self-BLEU が低いほど，多様性が高いことを示す．書き換えタスクではその書き換え出力において，prefix と suffix を保持しなければならない．この制約を満たした割合を Constraint として報告する．

1） 書き換えは単語単位で行うので，3 単語未満の文章は検証から除いた．
2） これらの設定はすべて，Evaluation by Huggingface のデフォルト・パラメーターを使用した．
3） https://openai.com/chatgpt
4） [12] のプロンプトを翻訳の評価に，[13] のプロンプトを要約の評価に使用し，各データセットからランダムに 100 件のサンプルを選択して評価した．

ベースライン: ベースラインモデルとして BART [5] と DElIteraTeR [4] を用いた．BART は書き換え候補を Text-Infilling タスクとして生成し，DElIteraTeR は書き換え候補を Paraphrase タスクとして生成する[5)]．これらのモデルは，この実験で使用されたデータセットとは異なるデータセットで学習されている．

復号手法: 貪欲な復号手法では，target と同じテキストが再度生成されるため，書き換え候補を生成する際には BS や確率的な復号手法を用いる必要がある．我々は DLBS と比較するためのベースラインとして Sample beam search (SBS) [15] と DBS を使って実験を行った．

ハイパーパラメータ: 我々は LA のパラメータを $\alpha_s = 0.5$，$\alpha_e = 0.1$ とし，DLBS のパラメータを $\beta = [0.8, 0.9, 1.0, 1.1, 1.2]$ とした．WJ のパラメータは要約タスクでは $d = 1e{-}3$ を，翻訳タスクでは $d = 1e{-}4$ を使用した．復号時にはビーム幅を $B = 10$ とし，サンプリングを上位 $k = 50$ の尤度に，グループサイズ $g = 5$ として実験を行った．

4 結果

XSum での実験結果 (表 1) は，SAGen が ChatGPT-Score において DElIteraTeR と同等であり，Self-BLEU において総合的に最も優れていることを示している．さらに，WMT19 の結果（表 1）では，ChatGPT-Score と Self-BLEU の両方で SAGen が最も優れている．BART と DElIteraTeR は，prefix と suffix を出力する制約に違反することが多く，DElIteraTeR の出力の半分は制約を満たしていなかった．

どちらのデータセットにも共通して，BART は出力の質を評価する ChatGPT-Score が他より低い傾向があった．これは，BART の言い換えた出力が極端に短くなる傾向から，その言い換え元のテキストとの乖離が大きくなったためである．DElIteraTeR は出力品質の指標は高いものの，多様性が低い．これはピリオドの追加や単語の削除などの小さな変更が多いためであり，このことから書き換えタスクには適さないとわかる．

Self-BLEU において SBS は DBS と比較して，XSum で 27.13 ポイント，WMT19 で 30.04 ポイント増加した．ここから，SBS は多様性の点で DBS より劣っているということがわかる．これは，SBS が確

5） DElIteraTeR は 5 つの編集注釈タグを使用して，どのように言い換えるかを指定する．タグの中で style が平均して良いスコアであったため，style を用いた出力を評価に用いる．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

	Decoding method	ChatGPT-Score				ROUGE 1/2/L	Self-BLEU	Constraint
		Rel.	Con.	Flu.	Coh.			
no-rewriting	-	8.53	9.60	8.21	8.47	43.82/21.76/36.29	-	-
BART	DBS	3.43	7.50	7.02	6.33	26.67/10.58/23.03	58.38	0.11
DElIteraTeR	DBS	7.72	9.32	**8.04**	**8.05**	35.07/15.03/29.50	61.86	0.50
SAGen	DBS	7.78	9.24	7.98	**8.05**	40.30/17.51/32.39	46.46	**1.00**
SAGen	SBS	**7.94**	**9.39**	7.94	8.04	**41.81/20.04/34.29**	73.29	**1.00**
	DLBS	7.77	9.20	7.96	8.02	40.31/17.52/32.40	**45.39**	**1.00**

	Decoding method	ChatGPT-Score	BLEU	Comet22	Self-BLEU	Constraint
no-rewriting	-	89.20	80.89	39.84	-	-
BART	DBS	32.61	57.46	8.38	57.50	0.65
DElIteraTeR	DBS	73.27	73.81	17.00	57.57	0.49
SAGen	DBS	81.46	**79.62**	29.77	48.42	**1.00**
SAGen	SBS	**82.69**	78.22	**38.92**	78.46	**1.00**
	DLBS	79.68	79.44	29.64	**47.71**	**1.00**

表 1 XSum での実験結果 (上) WMT19 での実験結果 (下). XSum の ChatGPT-Score は relevance, consistency, fluency, coherence の 4 つの観点で 10 段階で評価した。各指標で最も優れたスコアを **bold** で示した.

率的に次の候補を選択するため，スコアの高いビームの影響を強く受け，ほぼ同じテキストが複数出力されるためである．また，DLBS は DBS より XSum が 1.07 ポイント，WMT19 が 0.71 ポイント低かった．これは，DLBS が出力をさらに多様化する効果があることを示している．一方，品質に関する評価尺度である ChatGPT-Score は WMT19 では 1.78 ポイント低下したが，XSum はほぼ同じであった．ここからほとんどの場合 DBS を使用し，多様性がより重要な場合にのみ DLBS を使用することが適切であることがわかる．事例分析については Appendix 7.1 を参照されたい．

5 関連研究

Text infilling [16, 17, 18, 19] はテキストの欠落部分を生成するタスクである．Text infilling は特定の区間のテキストを書き直すと言う点で我々の提案する書き換えタスクと類似している．一方で，Text infilling は周囲の文脈のみを利用してテキストの欠落部分を生成するのに対し，我々のタスクは書き換え対象のテキストの特性を満たすように生成する．

周囲の文脈と明示的に結びついた文章を生成する手法として，[2] は，順方向と逆方向の 2 つの言語モデルを用いる手法を提案した．SAGen は復号手法を拡張することでこれを実現しており，追加のモデルの学習やデータセットを必要としない．

Paraphrasing は教師なし手法 [20, 21, 22, 23] と教師あり手法 [24, 25, 26, 3, 4] に分けられる．Paraphrase タスクは任意のテキストを書き換えるが，我々のタスクは書き換え対象が特定の言語モデルの出力であるため，その言語モデルに入力されたテキストも考慮した上で書き換えを行う違いがある．

復号手法は言語モデルがテキストを生成する仕組みである．復号手法の一つである BS は貪欲な復号手法より良いテキストを生成することが報告されている [27]．さらに BS を拡張する多くの手法が提案されている [28, 6, 29]．後方の文脈を考慮する復号手法として SAGen を提案し，また BS の拡張として長さを考慮する DLBS を提案した．

校正は高品質な文章を書くために不可欠なプロセスであり，人間と機械が分担して校正を行う取り組みが多くなされている [30, 5, 31, 1]．これらの研究では，人がすべての校正作業を行うよりも，機械と共同作業を行う方が効率的であることが報告されている．SAGen はこのような校正作業に利用できる．

6 まとめ

我々は言語モデルによって生成されたテキストの任意の区間を書き換えるタスクに取り組み，対象区間の書き換え候補を生成する手法 SAGen を提案した．SAGen の有効性を検証するために，要約と翻訳の 2 つのタスクで学習した言語モデルによって生成されたテキストに対して実験を行い，SAGen が品質と多様性の点でベースラインを上回ることを示した．SAGen は，モデルの学習やデータセットの作成を行うことなく，任意の言語モデルによって生成されたテキストを書き換えることができるため，多くの場面で校正に貢献できる．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

参考文献

[1] Wanyu Du, Zae Myung Kim, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. Read, revise, repeat: A system demonstration for human-in-the-loop iterative text revision. In **Proc. In2Writing**, pp. 96–108, 2022.

[2] Peter West, Ximing Lu, Ari Holtzman, Chandra Bhagavatula, Jena D. Hwang, and Yejin Choi. Reflective decoding: Beyond unidirectional generation with off-the-shelf language models. In **Proc. ACL-IJCNLP**, pp. 1435–1450, 2021.

[3] Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. Understanding iterative revision from human-written text. In **Proc. ACL**, pp. 3573–3590, 2022.

[4] Zae Myung Kim, Wanyu Du, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. Improving iterative text revision by learning where to edit from other revision tasks. In **Proc. EMNLP**, pp. 9986–9999, 2022.

[5] Mina Lee, Percy Liang, and Qian Yang. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In **Proc. CHI**, 2022.

[6] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. **CoRR**, Vol. abs/1610.02424, , 2016.

[7] Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. Facebook FAIR's WMT19 news translation task submission. In **Proc. WMT**, pp. 314–319, 2019.

[8] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In **Proc. EMNLP**, pp. 1797–1807, 2018.

[9] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Proc. Text Summarization Branches Out**, pp. 74–81, 2004.

[10] 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.

[11] Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In **Proc. WMT**, pp. 578–585, 2022.

[12] Mingqi Gao, Jie Ruan, Renliang Sun, Xunjian Yin, Shiping Yang, and Xiaojun Wan. Human-like summarization evaluation with chatgpt. **CoRR**, Vol. abs/2304.02554, , 2023.

[13] Tom Kocmi and Christian Federmann. Large language models are state-of-the-art evaluators of translation quality. In **Proc. EAMT**, pp. 193–203, 2023.

[14] Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In **Proc. SIGIR**, p. 1097–1100, 2018.

[15] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In **Proc. ACL**, pp. 889–898, 2018.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proc. NAACL**, pp. 4171–4186, 2019.

[17] Chris Donahue, Mina Lee, and Percy Liang. Enabling language models to fill in the blanks. In **Proc. ACL**, pp. 2492–2501, 2020.

[18] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. **Proc. TACL**, Vol. 8, pp. 64–77, 2020.

[19] 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 **Proc. ACL**, pp. 7871–7880, 2020.

[20] Dekang Lin and Patrick Pantel. Dirt @sbt@discovery of inference rules from text. In **Proc. KDD**, p. 323–328, 2001.

[21] Stanley Kok and Chris Brockett. Hitting the right paraphrases in good time. In **Proc. NAACL**, pp. 145–153, 2010.

[22] Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. Paraphrasing revisited with neural machine translation. In **Proc. EACL**, pp. 881–893, 2017.

[23] Aurko Roy and David Grangier. Unsupervised paraphrasing without translation. In **Proc. ACL**, pp. 6033–6039, 2019.

[24] Aaditya Prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Neural paraphrase generation with stacked residual LSTM networks. In **Proc. COLING**, pp. 2923–2934, 2016.

[25] Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. A deep generative framework for paraphrase generation. In **Proc. AAAI**, 2018.

[26] Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. FELIX: Flexible text editing through tagging and insertion. In **Find. EMNLP**, pp. 1244–1255, 2020.

[27] Markus Freitag and Yaser Al-Onaizan. Beam search strategies for neural machine translation. In **Proc. NGT Workshop**, pp. 56–60, August 2017.

[28] Chris Hokamp and Qun Liu. Lexically constrained decoding for sequence generation using grid beam search. In **Proc. ACL**, pp. 1535–1546, 2017.

[29] Tong Niu, Semih Yavuz, Yingbo Zhou, Nitish Shirish Keskar, Huan Wang, and Caiming Xiong. Unsupervised paraphrasing with pretrained language models. In **Proc. EMNLP**, pp. 5136–5150, 2021.

[30] Vishakh Padmakumar and He He. Machine-in-the-loop rewriting for creative image captioning. In **Proc. NAACL**, pp. 573–586, 2022.

[31] Nikhil Singh, Guillermo Bernal, Daria Savchenko, and Elena L. Glassman. Where to hide a stolen elephant: Leaps in creative writing with multimodal machine intelligence. **Proc. ACM Trans. Comput.-Hum. Interact.**, Vol. 30, No. 5, 2023.

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).

7 付録 (Appendix)

7.1 事例分析

各タスクの出力例を分析するために，モデルの尤度が最も高い上位 3 つのケースを示す．XSum の例 (表. 2) では，BART の出力はソース文の内容と乖離した文章を出力してしまっており，DElIteraTeR の出力は表面的な書き換えに過ぎない．一方，SAGen の最初の出力は，警察車両の窃盗に言及しておらず，3 番目の出力は，2 人の若者が「加重車両窃盗」で起訴されているという新しい情報を追加しているこの例から，SAGen の出力は内容レベルで多様な出力ができていると言える．

WMT19 の例 (表. 3) においても，BART の出力はソース文と乖離していた．DElIteraTeR の出力は書き換えとして問題ないが，文字 ”ß”を出力できていない．これは，DElIteraTeR が扱える語彙に ”ß”がないために生じたミスである．一方，SAGen は言い換え対象を生成したモデルを用いているため，正しく書き換えられている．これは，タスクごとに finetune されたモデルをそのまま利用する SAGen の優位性を示している．

モデル	出力
入力文書	Four police officers were injured in the incident on Friday night.A man, aged 19, and a boy, aged 16, have been charged with six counts of aggravated vehicle taking.They are due to appear before Belfast Magistrates' Court on Monday.The 19-year-old man has also been charged with driving while disqualified and using a motor vehicle without insurance.
正解の要約	Two teenagers have been charged in connection with an incident in west Belfast in which a car collided with two police vehicles.
no-rewriting	**Two teenagers** have been charged after a police car was stolen in **north Belfast.**
BART	**Two teenagers** shot dead in **north Belfast.** **Two teenagers** stabbed in **north Belfast.** **Two teenagers** stabbed to death in **north Belfast.**
DElIteraTeR	**Two teenagers** have been charged after a police car was stolen in **north Belfast.** **Two teenagers** have been charged after a police vehicle was stolen in **north Belfast.** **Two teenagers** have been charged in connection with the theft of a police car in **north Belfast.**
SAGen	**Two teenagers** have been charged in connection with an incident in **north Belfast.** **Two teenagers** have been charged in connection with an attempted carjacking in **north Belfast.** **Two teenagers** have been charged with aggravated vehicle taking following an incident involving a police car in **north Belfast.**

表 2 Xsum の出力例．prefix と suffix を **bold** で示している．

モデル	出力
ドイツ語の原文	Ja, hier an der Auerfeldstraße sperrte man einst Menschen mit Wahnvorstellungen ein.
正解の翻訳文	Yes, here on Auerfeldstraße, they once imprisoned people with delusions.
no-rewriting	**Yes,** people with delusions were once imprisoned here on **Auerfeldstraße.**
BART	**Yes,** people still live on **Auerfeldstraße.** **Yes,** people are still on **Auerfeldstraße.** **Yes,** people live on **Auerfeldstraße.**
DElIteraTeR	**Yes,** people with delusions were once imprisoned on Auerfeldstrae. **Yes,** people with delusions were imprisoned on Auerfeldstrae. **Yes,** people are imprisoned on Auerfeldstrae.
SAGen	**Yes,** people with delusions were once imprisoned here on **Auerfeldstraße.** **Yes,** people with delusions were once locked up here on **Auerfeldstraße.** **Yes,** people who had delusions were once imprisoned here on **Auerfeldstraße.**

表 3 WMT19 の出力例．prefix と suffix を **bold** で示している．

This work is licensed by the author(s) under CC BY 4.0
(https://creativecommons.org/licenses/by/4.0/).