

近傍の事例を用いた非自己回帰生成

丹羽彩奈 高瀬翔 岡崎直観
東京工業大学 情報理工学院

{ayana.niwa [at] nlp., sho.takase [at] nlp., okazaki [at]} c.titech.ac.jp

概要

非自己回帰モデルは、自己回帰モデルよりも少ない計算量で文を生成できるが、生成文の品質が劣るため、生成を反復する等の工夫が要る。本稿では、編集操作に基づく非自己回帰モデルの初期状態に入力文の近傍の事例を活用し、反復回数を抑えつつ生成品質を改善する手法を提案する。また、初期事例の選び方で性能が変化すること、理想的な事例を選択できれば、非自己回帰モデル単体でも自己回帰モデルの性能を上回る可能性があることを報告する。

1 はじめに

非自己回帰モデル [1] は、近年活発に研究が進められているテキスト生成の枠組みのひとつである。自己回帰モデル [2] では、デコーダの入力層から出力層への順伝播の計算を生成されるトークンの数だけ繰り返す必要があるが、非自己回帰モデルは複数のトークンを同時に予測するため、順伝播の計算が1回で済み、生成が速い。一方で、生成されるトークン間の依存関係を十分に捉えられず、自己回帰モデルに比べて生成文の品質が劣る傾向にある。

これに対し、非自己回帰モデルの生成を複数回繰り返すことで生成文の質を向上させる手法が提案されているが、生成速度の優位性を維持したまま自己回帰モデルと同等の生成品質に引き上げることは達成できていない [3, 4]。また品質向上のために自己回帰モデルを利用した知識蒸留 [5] がほぼ必須である。

本研究では、非自己回帰モデルによる生成文の品質をより少ない回数の生成で向上させるため、近傍の事例を初期状態に用いる手法を提案する。近傍の事例は、品詞タグ付け [6] や、用例ベース機械翻訳 [7, 8] など、幅広いタスクで活用されてきた。近年では、特にニューラル機械翻訳などのタスクで自己回帰モデルにおける近傍の事例の有効性が報告されている [9, 10, 11]。しかし、非自己回帰モデルにおける近傍の事例の有効性は検証されていない。

提案手法では、Levenshtein Transformer [12] をベースに、検索された近傍の事例に対してトークンの挿入・削除を行い、文を生成する。例えば、“I have an apple.”という英文を“*Ich habe einen Apfel.*”という独文に翻訳する時、“I have a banana.”—“*Ich habe eine Banane.*”という英独の翻訳対を近傍の事例として見つけ、その独文を初期状態として‘eine’, ‘Banane’という単語を削除し、‘einen’, ‘Apfel’という単語を挿入することで、正しい独訳を生成できる。これにより、全てのトークンを無から生成するよりも容易に文を生成できるため、品質の向上と生成回数の削減を期待できる。検索された近傍の事例からターゲット文への編集操作を効率的に学習するため、本研究では近傍の事例とターゲット文の違いに着目した学習方策を設計する。既存研究 [13, 14] は近傍の事例をエンコードする特別な機構やパラメータが必要であったが、提案手法は学習法策の変更により近傍の事例を考慮したデコーディングを実現する。

評価実験では、非自己回帰モデルにおいても近傍の事例は有用であり、既存の非自己回帰モデルと比べ少ない反復回数で高い生成品質を達成できることが分かった。また、理想的な近傍の事例を選ぶことができれば、非自己回帰モデル単体でも自己回帰モデルの性能を超える可能性があることも報告する。

2 提案手法

本研究では、ソース文 x と近傍の事例 z_0 を用いてターゲット文 y^* を生成する系列変換タスクに取り組み。図 1 に示す通り、提案手法は翻訳メモリ (学習事例の全文対) から近傍の事例を検索する (2.2 節)。次に、検索した近傍の事例に対してトークンの削除・挿入を行うことで文を生成する (2.1 節)。

2.1 近傍の事例の編集モデル

モデルの概要 提案モデルは、編集に基づく非自己回帰モデル Levenshtein Transformer [12] に着想を得た系列変換モデルである。デコーダは、不要

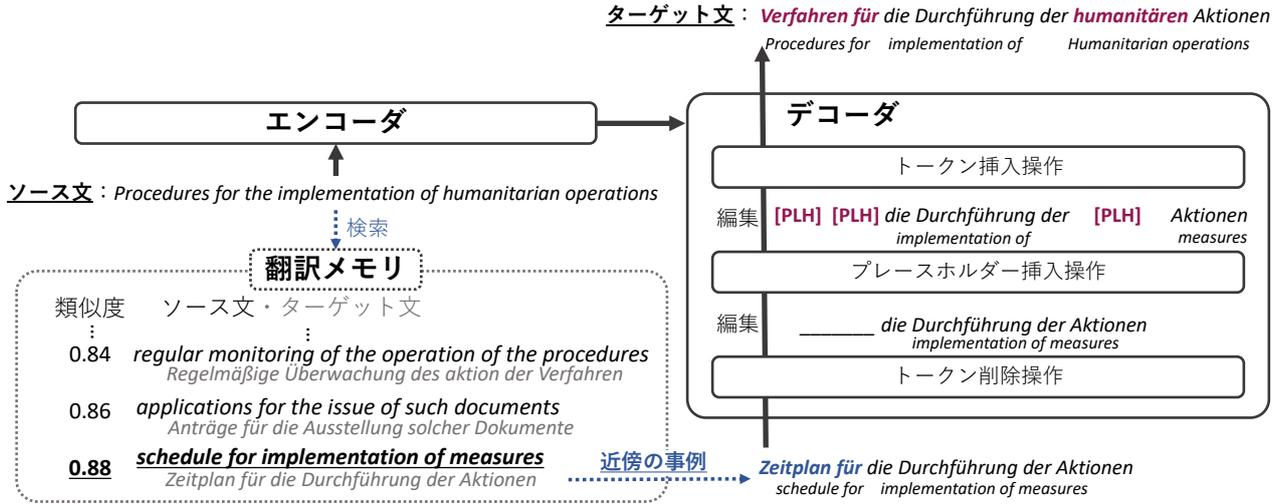


図 1 提案手法の概要. 英文 “Procedures for the implementation of humanitarian operations” を独文 “Verfahren für die Durchführung der humanitären Aktionen” に翻訳するとき, 近傍の事例 “Zeitplan für die Durchführung der Aktionen (schedule for implementation of measures)” を初期状態として編集を繰り返す.

なトークンの削除, プレースホルダー [PLH] の挿入, [PLH] への単語の挿入の機構を持つ. 近傍の事例 z_0 からターゲット文 y^* への編集操作は, 模倣学習で学習する. 模倣学習は強化学習の一種で, エキスパートの一連の行動を模倣し, 同様の行動ができるように学習する. モデルはステップ k で前ステップで編集済みの系列 y^{k-1} を受け取り, 次の系列 $y^k = \mathcal{G}(y^{k-1}, a^k)$ を生成するための行動 (編集操作) $a^k \in \mathcal{A}$ を選択する. ここで, \mathcal{G} は編集操作と系列を受け取り, 編集済みの系列を返す環境である. この行動を選択するための方策 π は, 入力系列 $y_{0:n}^{k-1}$ に対するデコーダの出力 (h_0, h_1, \dots, h_n) から行動空間 \mathcal{A} への写像として定義される. 方策 π は線形分類器で近似するのが一般的であり, 本モデルにおける方策 π は削除すべきトークン位置の予測器 π^{del} , [PLH] を挿入すべき位置と数の予測器 π^{plh} , そして [PLH] に挿入すべきトークンの分類器 π^{tok} からなる.

学習時 方策分類器の学習では, 系列 y からエキスパートとなる文 e を生成するための正しい行動を示すオラクル方策 π^* を設計する. モデルは, その系列 y とエキスパート e の編集距離 \mathcal{D} が最も小さくなるように編集操作を選択する. なお, 本模倣学習における報酬関数はこの編集距離 \mathcal{D} の負の値をとったものである.

$$a^* = \arg \min_a \mathcal{D}(\mathcal{G}(y, a), e) \quad (1)$$

そして, 入力事例に対する最適な編集操作 a^* の確率が最大となるようにモデルの方策 $\pi_\theta = \pi^{\text{del}} \cdot \pi^{\text{plh}} \cdot \pi^{\text{tok}}$ を更新する. 本研究では, 近傍の事例に対する編集

操作を効率的に学習するため, 各編集操作におけるオラクル方策 π^* を次のように設計した.

1. **削除操作**: 近傍の事例に対する編集操作において重要なのは, 近傍の事例内に含まれる不要な単語を削除することである. そこで, 近傍の事例 z_0 を削除操作の対象 y_{del} とし, ターゲット文 y^* に存在しない単語を削除する学習を行う.

$$y_{\text{del}} = z_0 \quad (2)$$

$$\pi^*(y_{\text{del}}) = C(y^*, z_0) \quad (3)$$

ここで, $C(y^*, z_0)$ は系列 y^* と z_0 の共通単語列を返す関数である.

2. **挿入操作**: ターゲット文をエキスパートとし, トークンが削除された文に必要なトークンを挿入する学習を行う. 挿入操作の対象となる系列 y_{ins} にはモデルが系列 y に削除操作 \vec{d} を行った後の不完全な事例 $\mathcal{G}(z_0, \vec{d})$ とターゲット文 y^* からランダムに単語を削除した事例 $\mathcal{G}(y^*, \vec{d})$ のいずれかからランダムに選択し, 挿入操作を学習する.

$$y_{\text{ins}} = \begin{cases} \mathcal{G}(y^*, \vec{d}), \vec{d} \sim \pi^{\text{RND}} & (\text{if } u < \alpha) \\ \mathcal{G}(z_0, \vec{d}), \vec{d} \sim \pi_\theta & (\text{otherwise}) \end{cases} \quad (4)$$

$$\pi^*(y_{\text{ins}}) = y^* \quad (5)$$

ここで, π^{RND} はランダムに単語を削除する方策, π_θ は学習中のモデルの方策, $\alpha \in [0, 1]$ はハイパーパラメータ, u は $[0, 1]$ から一様分布に従ってサンプリングした値である. 挿入操作は, [PLH] が必要な位置とその数を予測した後, [PLH] を埋めるトークンを予測するという二つのサブタスクからなる.

推論時 事前に検索した近傍の事例 z_0 を最初のデコーダへの入力として、最も高確率な編集操作を選択・適用することを繰り返すことで文を生成する。このとき、反復回数が上限に達するか、同じ編集操作が予測された場合に編集を終了させる。

2.2 近傍の事例の検索

提案手法では、学習データ内のすべての文対を含む翻訳メモリから生成したい文に最も類似した事例を検索し、学習時・推論時の編集対象とする。既存の近傍の事例の検索方法は、語彙マッチング [9, 11] とベクトル [10, 14] の二種類に大別される [9]。

語彙マッチング いくつかの方法が考えられるが、今回は大きなデータにもスケールするように、近傍の事例を用いた既存研究 [15] と同様に、単語の出現頻度をベクトル化する。具体的には、TFIDF ベクトル $\text{tfidf}(\cdot)$ のコサイン類似度を用いて事例 s_i, s_j の近さスコア S_{TFIDF} を定義する。

$$S_{\text{TFIDF}}(s_i, s_j) = \frac{\text{tfidf}(s_i) \cdot \text{tfidf}(s_j)}{\|\text{tfidf}(s_i)\| \|\text{tfidf}(s_j)\|} \quad (6)$$

また、より大域的な系列の類似度を考慮するため、TFIDF ベクトルの類似度が上位 n 件の事例を最長共通部分文字列 (LCS) の長さでランキングする設定でも実験を行う (TFIDF+LCS)。

文ベクトル エンコーダから計算される文ベクトルのコサイン類似度 S_{SentVec} を用いる。

$$S_{\text{SentVec}}(s_i, s_j) = \frac{\mathbf{h}_i \cdot \mathbf{h}_j}{\|\mathbf{h}_i\| \|\mathbf{h}_j\|} \quad (7)$$

ここで、 $\mathbf{h}_i, \mathbf{h}_j$ はそれぞれ、文 s_i, s_j をエンコードして得られた文ベクトルである。

3 実験

機械翻訳タスクを用いて、非自己回帰生成モデルにおける近傍の事例の効果を調べる。

3.1 実験設定

データセット 本研究では法文書コーパスである JRC-Acquis¹⁾ を利用する。本実験では、重複する対訳対の事例を除外した後、Gu ら [11] と同様の前処理を行なった。各言語対の事例数を表 1 に示す。トークナイザには Moses Tokenizer²⁾ を用いた。

モデル 自己回帰モデルの (AR) ベースラインとして Transformer [16] の base 設定を用いる。また、

表 1 JRC-Acquis コーパスに含まれる対訳文の数

	英独	英仏	英西
訓練データ	475,330	466,704	467,275
開発データ	2,945	2,935	2,922
評価データ	2,937	2,903	2,916

編集操作に基づく非自己回帰モデル (NAR) のベースラインとして Levenshtein Transformer (Levenshtein Trans) [12] を用いる。提案手法は NeighborEdit と表記する。モデルは fairseq [17]³⁾ をベースに実装した。学習は全て 1 ステップあたりのバッチサイズを 32,768、ステップ数は 40,000 回とした。

近傍の事例の検索法 2.2 節で説明した TFIDF, SentVec で翻訳メモリを検索し、入力に最も近い事例を選択する。TFIDF+LCS では、TFIDF で検索された上位 10 件を LCS の長さでランキングした。比較のため、訓練データからランダムに選んだ場合も実験した。文ベクトルのエンコーダには多言語コーパスで事前学習済みの SBERT モデル⁴⁾ [18] を、近傍検索には faiss⁵⁾ を用いた。

評価 生成の品質評価では、SacreBLEU [19] を用い、大文字小文字を区別する BLEU で比較する。また、一事例あたりの推論に要した平均反復回数と平均所要時間を測定する。後者は、先行研究 [20] に従い、1 個の GPU を用いてバッチサイズ 1 で文を生成し、一事例を生成し終えるまでの所要時間の平均値を求めた。なお、提案手法の平均所要時間には近傍の事例の検索にかかる時間も含めた。

3.2 実験結果

表 2 に BLEU, 平均反復回数, および所要時間 [ms] を示した。また、提案手法とベースラインの出力例を A.1 に示した。近傍の事例を初期状態に用いる NeighborEdit は、非自己回帰モデルのベースライン (Levenshtein Trans) の性能を 2.2 から 6.3 ポイントほど上回った。平均反復回数もベースラインから最大で 25%以上減少した。この結果は、近傍の事例を非自己回帰モデルに組み込むことで、品質を向上させながら反復回数を削減できることを示している。さらに、推論にかかる所要時間もほぼ全ての設定でベースラインよりも短くなった。この所要時間に関しては、付録 A.2 で補足した。

3) <https://github.com/pytorch/fairseq>

4) <https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1>

5) <https://github.com/facebookresearch/faiss>

1) <https://opus.nlpl.eu/JRC-Acquis.php>

2) <https://github.com/moses-smt/mosesdecoder/>

表 2 JRC-Acquis データセットでの BLEU スコアと平均反復回数と平均所要時間 (太字は非自己回帰モデルでの最良値)

モデル	デコーダの初期状態	BLEU			平均反復回数			平均所用時間 [ms]		
		英独	英仏	英西	英独	英仏	英西	英独	英仏	英西
Transformer (AR)	[BOS]	53.2	64.2	61.0	23.21	27.49	29.26	300.8	321.7	330.7
Levenshtein Trans (NAR)	<s></s>	45.4	57.8	55.5	2.48	2.21	2.27	80.8	72.2	73.7
NeighborEdit (NAR)	TFIDF	51.3	60.6	58.6	1.97	1.67	1.99	79.2	67.3	76.5
	TFIDF+LCS	51.6	62.2	58.5	1.92	1.69	1.92	76.5	68.4	74.1
	SentVec	51.4	63.0	57.7	1.95	1.65	1.81	78.9	66.0	71.4
	ランダム	25.6	45.5	45.6	3.88	3.63	3.50	131.0	123.0	120.4

表 3 SRC と ORACLE の BLEU スコアの比較 (英独, * は自己回帰モデルの BLEU スコア 53.22 を超えた数値を指す)

近傍の 検索法	学習時の クエリ	推論時のクエリ		
		SRC	ORACLE	ランダム
TFIDF	SRC	51.26	52.95	34.05
	ORACLE	51.21	55.25*	27.31
TFIDF +LCS	SRC	51.63	53.98*	32.22
	ORACLE	52.19	56.76*	27.41
SentVec	SRC	51.43	51.91	35.56
	ORACLE	51.75	52.67	32.76

続いて、提案手法の初期系列 z_0 が生成の質に与える影響を詳しく調べるため、学習時と推論時の検索クエリにそれぞれソース文 (SRC) とターゲット文 (ORACLE) を用いた場合を比較する。後者は、理想的な近傍の事例を選択した場合の上限性能を表す。表 3 に英独での結果を示した通り、特に語彙マッチングを用いた検索では、推論時に ORACLE を用いることで BLEU スコアが向上する (+1.69 から +4.56 ポイント)。一方で、学習時に ORACLE を組み込んでも、BLEU スコアはさほど変化しないため、提案手法の性能向上のためには、編集操作の訓練データを理想に近づけるよりも、初期状態を改善することが重要である。つまり、適切な近傍の事例を選択できれば、提案手法は自己回帰モデル (AR) の性能を超える可能性がある。これは、英仏・英西でも同様であった (付録 A.3 参照)。しかし、現実の問題設定では ORACLE のターゲット文を用いることができない。

そこで、SRC で検索した事例のみで ORACLE の性能に迫ることができるかを調べるため、用いる近傍の事例を 5 件に増やし、それぞれ並列に文を生成し、その中で最も BLEU の値が高い事例を出力文と見なした場合の結果を表 4 に示した。その結果、すべての検索法で大幅に性能が向上し (+4.60 から +5.45 ポイント)、ORACLE (上位 1 件) と同等以上の性能を達成できた。つまり、SRC をクエリとした場合でも、

表 4 近傍の事例数を変えた場合の BLEU スコア (英独)

近傍の 検索法	SRC		ORACLE	
	1 件	5 件	1 件	5 件
TFIDF	51.26	56.08	52.95	56.63
TFIDF+LCS	51.63	56.23	53.98	57.09
SentVec	51.43	56.88	51.91	56.73

リランキングによって自己回帰モデルの性能を英独翻訳で 2.88 ポイント以上上回る可能性がある。

しかし、提案手法には改善の余地がある。例えば、非自己回帰モデルによく見られる問題である単語の反復 [1] がベースラインよりも頻繁に見られる。これは、提案手法の方策がモデルの出力に対する削除操作を対象としておらず、デコードの途中で余計に生成された単語を反復の過程で削除しにくいことである。また、近傍の事例から出力文を生成するまでに、複数回の反復を要する事例でも生成文の品質が低下する傾向が見られる。単語の重複を削除したり、複数の反復にわたって編集を行う場合の性能を改善するため、モデルの出力に対する編集操作を組み込んだ方策の設計を今後の課題としたい。

4 おわりに

本稿では、編集操作に基づく非自己回帰モデルに近傍の事例を組み込む手法を提案し、既存手法よりも少ない反復回数でも生成文の質を大幅に改善できることを報告した。また、近傍の事例の選び方で性能が大きく変化すること、近傍の事例を適切に選択できれば、非自己回帰モデル単体でも自己回帰モデルの性能を超える可能性があることを示した。今後の展望としては、より良い文単位での近傍の事例の検索法の検討やモデルの出力に対して繰り返し編集を行うための方策の設計、複数の近傍の事例を同時に考慮するモデルの開発を行いたい。

謝辞

本研究は JSPS 科研費 21J13602 の助成を受けたものです。

参考文献

- [1] Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. Non-autoregressive neural machine translation. In **International Conference on Learning Representations**, 2018.
- [2] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In **Advances in Neural Information Processing Systems**, Vol. 27, 2014.
- [3] Jason Lee, Raphael Shu, and Kyunghyun Cho. Iterative refinement in the continuous space for non-autoregressive neural machine translation. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1006–1015, Online, November 2020. Association for Computational Linguistics.
- [4] Yinhan Liu, Luke Zettlemoyer, Marjan Ghazvininejad, Omer Levy. Mask-predict: Parallel decoding of conditional masked language models. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing**, 2019.
- [5] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In **Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing**, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics.
- [6] Walter Daelemans, Jakob Zavrel, Peter Berck, and Steven Gillis. MBT: A memory-based part of speech tagger-generator. In **Fourth Workshop on Very Large Corpora**, Herstmonceux Castle, Sussex, UK, June 1996. Association for Computational Linguistics.
- [7] Makoto Nagao. A framework of a mechanical translation between Japanese and English by analogy principle. In **Proceedings of the International NATO Symposium on Artificial and Human Intelligence**, pp. 173–180, 1984.
- [8] Eiichiro Sumita and Hitoshi Iida. Experiments and prospects of example-based machine translation. In **29th Annual Meeting of the Association for Computational Linguistics**, pp. 185–192, Berkeley, California, USA, June 1991. Association for Computational Linguistics.
- [9] Jitao Xu, Josep Crego, and Jean Senellart. Boosting neural machine translation with similar translations. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 1580–1590, Online, July 2020. Association for Computational Linguistics.
- [10] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In **International Conference on Learning Representations**, 2021.
- [11] Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. Search engine guided neural machine translation. In **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 32, 2018.
- [12] 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, **Advances in Neural Information Processing Systems**, Vol. 32. Curran Associates, Inc., 2019.
- [13] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Retrieve, rerank and rewrite: Soft template based neural summarization. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 152–161, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [14] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens, 2021.
- [15] Hao Peng, Ankur Parikh, Manaal Faruqui, Bhuwan Dhingra, and Dipanjan Das. Text generation with exemplar-based adaptive decoding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 2555–2565, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Advances in neural information processing systems**, pp. 5998–6008, 2017.
- [17] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)**, pp. 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [18] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [19] Matt Post. A call for clarity in reporting bleu scores. **WMT 2018**, p. 186, 2018.
- [20] Weijia Xu and Marine Carpuat. EDITOR: An edit-based transformer with repositioning for neural machine translation with soft lexical constraints. **Transactions of the Association for Computational Linguistics**, Vol. 9, pp. 311–328, 2021.

ソース文	(48) Capacity utilisation increased by 56 % between 1996 and the IP .
ターゲット文	(48) Die Kapazitätsauslastung stieg von 1996 bis zum UZ um 56 % .
Levenshtein Transformer	
削除操作 1	
挿入操作 1	[PLH] [PLH] (48) Die Kapazitätsauslastung stieg von 1996 bis dem UZ UZ um 56 % .
削除操作 2	(48) Die Kapazitätsauslastung stieg von 1996 bis dem UZ um 56 % .
挿入操作 2	(48) Die Kapazitätsauslastung stieg von 1996 bis [PLH] dem UZ um 56 % . (48) Die Kapazitätsauslastung stieg von 1996 bis zu dem UZ um 56 % .
NeighborEdit	
近傍の事例 (英)	(56) Export prices of the two cooperating Thai producers have increased by 6 % between 1996 and the IP .
近傍の事例 (独)	(56) Die Ausführpreise der beiden kooperierenden thailand@ischen Hersteller stiegen von 1996 bis zum Untersuchungszeitraum um 6 %
削除操作 1	() Die von 1996 bis zum um % . ([PLH]) Die [PLH] [PLH] von 1996 bis zum [PLH] um [PLH] % .
挿入操作 1	(48) Die Kapazitätsauslastung stieg von 1996 bis zum UZ um 56 % .

図 2 実際の出力例。(英独, 赤文字が挿入されたトークンで, 青文字が削除されたトークン)

表 5 SRC と ORACLE の BLEU スコアの比較 (英仏, * は自己回帰モデルの BLEU スコア 64.21 を超えた数値)

近傍の 検索法	学習時の クエリ	推論時のクエリ		
		SRC	ORACLE	ランダム
TFIDF	SRC	60.55	62.27	42.36
	ORACLE	60.04	63.29	35.23
TFIDF +LCS	SRC	62.24	64.66*	44.92
	ORACLE	61.16	65.59*	38.02
SentVec	SRC	62.96	63.59	47.05
	ORACLE	62.84	64.19	42.96

表 6 SRC と ORACLE の BLEU スコアの比較 (英西, * は自己回帰モデルの BLEU スコア 61.01 を超えた数値)

近傍の 検索法	学習時の クエリ	推論時のクエリ		
		SRC	ORACLE	ランダム
TFIDF	SRC	58.57	59.79	45.00
	ORACLE	58.41	61.13*	39.10
TFIDF +LCS	SRC	58.53	60.64	45.49
	ORACLE	59.16	63.13*	40.18
SentVec	SRC	57.72	58.46	42.94
	ORACLE	59.03	59.87	46.28

A 付録 (Appendix)

A.1 モデルの出力例

ベースライン (Levenshtein Transformer) と提案手法 (NeighborEdit) の実際の出力例を図 2 に示した。提案手法は、近傍の事例から不要な単語を削除した結果をテンプレートのように入れて必要な単語を穴埋めすることで、より少ない反復回数で正解文を生成することができる。

A.2 推論時の所要時間

表 2 で示したように、提案手法は平均反復回数がベースラインと比較して少ないが、一方で推論に要する所要時間の差は顕著でない。これは、提案手法が近傍の事例の検索時間を含めていることよりも、Levenshtein Transformer が 1 反復目の削除操作をスキップすることが原因として考えられる。なお、提案手法の近傍の事例一件あたりの平均検索時間は、TFIDF (Top-1 の検索) で 0.11 [ms], TFIDF+LCS (Top-10 の検索) で 0.13 [ms], SentVec (Top-1 の検索) で 0.08 [ms] であった。

A.3 SRC と ORACLE の BLEU スコアの比較 (英仏, 英西)

英仏, 英西のデータセットを用いた検索のクエリ SRC と ORACLE の比較を表 5, 6 に示した。これらのデータセットにおいても、訓練データの質よりも推論時のデコーダの初期値の質を向上させることで性能改善が見られた。また、推論時の近傍の事例を適切に選択できれば自己回帰生成の性能を超えられる場合がある。