テキスト編集事例の編集操作への自動分解

山口大地[1] 宮田玲[2] 藤田篤[3] 梶原智之[4] 佐藤理史[1]
[1] 名古屋大学大学院工学研究科 [2] 東京大学大学院教育学研究科
[3] 情報通信研究機構 [4] 愛媛大学大学院理工学研究科
yamaguchi.daichi.e4@s.mail.nagoya-u.ac.jp

概要

テキスト編集システムの能力を評価するためには、システムが適用した編集操作を基本的な単位で同定することが重要である。このような編集操作の同定に向け、本稿ではテキスト編集事例の分解タスクを、所与の原文と編集文の間をつなぐ、非冗長かつ文法性を保つ、意味的に関連した最小のまとまりであるプリミティブな編集操作の系列を生成するタスクと定義する。これを自動化するためにアライメントツールと事前学習済み大規模言語モデルを用いた手法を提案し、テキスト平易化と機械翻訳後編集のデータセットを用いた実験の結果について報告する。

1 はじめに

テキスト編集は人間の執筆活動において重要な役割を担う。その自動化はテキスト平易化や文法誤り訂正、機械翻訳向け前編集・後編集、文圧縮など様々な自然言語処理タスクにおいて取り組まれてきた。これらの性能は、深層学習に基づくテキスト編集技術により大きく向上してきた [1, 2, 3, 4]。

テキスト編集システムの振る舞いを理解することは、システムの編集能力の理解につながるため、重要である。深層学習に基づくテキスト編集システムを開発・評価する際の問題点として、システムの振る舞い、つまり、システムが編集文を生成するためにどのように原文を編集したか、を説明できないことが挙げられる。システムの評価には、BLEU [5] や ROUGE [6]、BLEURT [7] などの評価指標が広く用いられている。しかし、これらの評価指標はいずれもシステムの振る舞いを一つの数値に集約するものであるため、解釈可能性が低い [8]。

システムの振る舞いを理解する方法の 1 つに分析的評価がある [9]。この評価手法は詳細な分析を可能にするものの、他の人手評価と比較して、労力がかかる。従って、システム開発の高速なサイクルに取り入れるためには、評価の自動化が求められる。

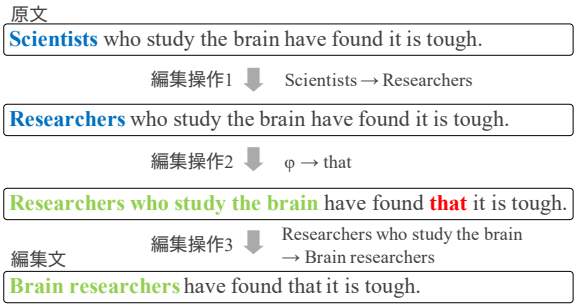

図 1 テキスト編集事例の編集操作への分解

分析的評価は原文・編集文の対における編集操作の同定と各編集操作の分類という 2 つのサブタスクで構成される。編集操作は、所与の文対を文法性を保つ最小単位の編集操作に分解することで同定できる [10]。しかし、これまでの研究において編集操作の同定はアノテータの経験に基づいて行われており、形式化も自動化もされておらず、アノテータ間での一貫した編集操作の同定の難しさが示唆されている [11]。一方、編集操作の分類については、様々なタスクに対して編集操作の分類体系が構築されており [9, 12, 13, 14]、自動分類も試みられている [15]。

本稿ではテキスト編集事例の分解タスクの形式化とその自動化の試みについて述べる。一般に、人間によるテキスト編集は文法性を保ち、意味的に関連した最小のまとまりであるプリミティブな編集操作を組み合わせることで行われる。例えば、図 1 に示す原文と編集文の対は 3 つの独立したプリミティブな編集操作に分解できる。このような分解の自動化に向けて、アライメントツールと事前学習済み大規模言語モデル (LLM) を用いた手法を実装し、テキスト平易化と機械翻訳後編集のデータセットを用いて有効性を検証した。その結果、それぞれのデータセットに含まれる編集事例の 44%と 64%を適切な粒度と順序の編集操作系列に分解できた。

2 問題設定

2.1 タスクの定式化

所与の原文と編集文の対 (X_{src}, X_{edt}) に対して、その文対をつなぐ編集操作系列を生成する。一般に、系列の順序は半順序となるため、分解結果は編集操作のラティスになる。形式的には、以下のように表すことができる。

$$\mathbb{E} = \{\mathbf{E} = (E_1, \ldots, E_n) \mid X_{edt} = E_n(\ldots E_1(X_{src}))\}$$

ここで、$\mathbf{E}$ は 2.3 節で述べる制約を満たす編集操作系列である。E_i は個々の編集操作を表す関数であり、引数の入力文 X_{i-1} に対して、あるスパンを別の表現に変換した X_i $(= E_i(X_{i-1}))$ を返す。添字の i は編集操作の適用順序を表す。以降では、X_{i-1} 内の表現 A を他の表現 B に変換する編集操作 E_i を "A ⇒ B" と表し、A を原文側、B を編集文側と呼ぶ。

本タスクでは、あらゆる編集操作系列を網羅するラティスの生成を目指すのではなく、半順序を満たす妥当な編集操作系列を 1 つ生成することを目指す。これは、システムの振る舞いを理解するためには、1 つの妥当な系列で十分なためである。

2.2 タスクの特徴

本タスクには、考慮すべき特徴が 3 つある。

a. 編集操作の順序　編集操作間に順序の制約が存在する場合がある。例えば、以下の (X_{src}, X_{edt}) は "me ⇒ ϕ " と "tell ⇒ state" の 2 つの編集操作で説明できる。しかし、state は目的語を 1 つしか取らないため、後者は前者の適用後にのみ適用できる。

X_{src}: Please **tell** **me** the truth.
X_{edt}: Please **state** the truth.

b. 表層上に現れない編集操作　編集操作の原文側あるいは編集文側に、X_{src} と X_{edt} のどちらにも現れない語が含まれる場合がある。以下の例を考える。

X_{src}: He **can** **use** that.
X_{edt}: He **does** that.

この (X_{src}, X_{edt}) は以下のように、2 つの編集操作を用いて、2 通りの説明ができ、それぞれ "do" と "uses" が X_{src} にも X_{edt} にも現れない。

E_1: "use ⇒ **do**",　　E_2: "can **do** ⇒ does"
E_1: "can use ⇒ **uses**",　E_2: "**uses** ⇒ does"

c. 不連続なスパンの編集操作　編集操作の原文側あるいは編集文側が不連続なスパンの語で構成される場合がある。例えば、編集操作 "like ... very much ⇒ love" の原文側は不連続なスパンである。

2.3 制約

編集操作、および編集操作系列は以下の 3 つの制約を満たすものとする。

文法性　編集操作が適用された後のスパンは文法誤りを含まない。人間が理解できる編集操作として、文法性を保つ操作であることを要件とする。

非逸脱性　所与の原文と編集文の対に対して、編集操作系列は冗長な編集操作を含まない。

プリミティブ性　編集操作の原文側と編集文側がそれぞれ意味的に関連した最小のまとまりであることを、編集操作がプリミティブであると定義する。この制約を考慮することで、一貫した単位での編集操作の同定が可能になる。編集操作のプリミティブ性は原文側と編集文側のどちらにも依存する。以下の例を考える $(i < j)$。

X_{i-1}:　I **like** David Bowie very much, **too**.
X_i　:　I **love** David Bowie very much, **too**.
X_j　:　I **love** David Bowie very much.

この組では、"like ⇒ love" はプリミティブである。しかし、以下の例では "like ... very much" と "love" が意味的に関連した最小のまとまりであるため、上の例と同じ編集操作 "like ⇒ love" はプリミティブではない。このように、プリミティブ性の判定は X_j に依存して異なりうる。

X_{i-1}:　　I **like** David Bowie **very** **much**, **too**.
X_i　:　* I **love** David Bowie very much, **too**.
X_j　:　　I **love** David Bowie.

3 提案手法

所与の原文と編集文の対に対する編集操作系列を生成するために、編集過程の文をノード、編集操作をエッジとするラティスを生成し、その中の 1 つのパスを選択する。本タスクでは、ラティスの始端ノードと終端ノードが与えられており、ノードとエッジは少数であるため、全探索が可能である。

3.1 ラティス生成

$s < t$ なる任意の文対 (X_s, X_t) に対する中間文候補の生成、フィルタリングを繰り返すことでラティスを生成する (図 2)。ここで、文法性と非逸脱性を満たす編集操作で生成された文を (X_s, X_t) に対す

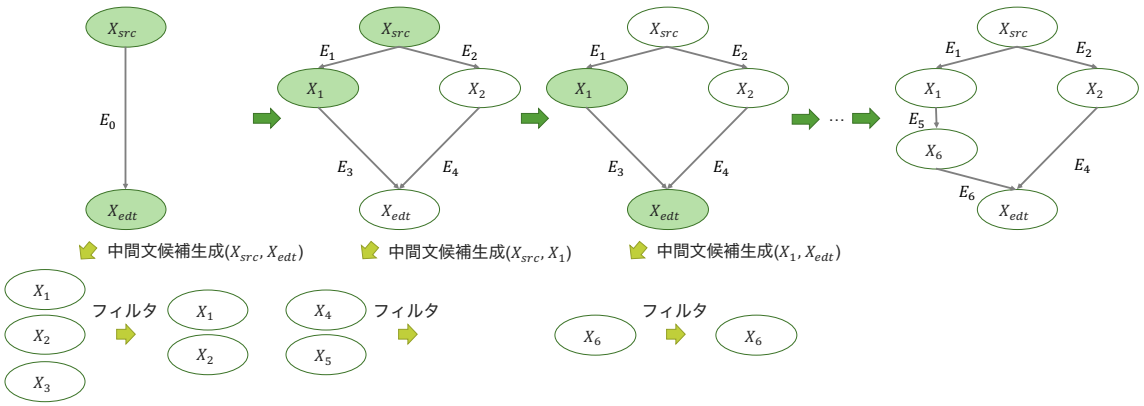

図 2　編集操作のラティスの生成手順

る中間文 X_{inter} $(s < inter < t)$ と定義する。まず、ラティスを (X_{src}, X_{edt}) で初期化する。次に、中間文候補生成器を用いて、(X_{src}, X_{edt}) に対する中間文候補集合 $\mathbb{C}$ を生成し、中間文フィルタを用いて中間文集合 $\mathbb{I} \subseteq \mathbb{C}$ を抽出する。そして、$\mathbb{I} \neq \phi$ の場合は (X_{src}, X_{edt}) のエッジを削除し、$\forall X \in \mathbb{I}$ について (X_{src}, X) と (X, X_{edt}) のエッジを追加する。ラティス中の全てのエッジに対して、新たな中間文が生成できなくなるまで同様のプロセスを再帰的に適用することで、ラティスを得る。

3.1.1　中間文候補生成器

中間文候補の生成には、以下の手法を用いる。

アライメント手法　所与の文対に対して、フレーズアライメントを同定し、得られた複数のアライメントの各々を独立した編集操作とする。これらを原文側に適用することで複数の中間文候補を生成する。この手法では、2.2 節で説明した b と c の特徴を持つ編集操作で生成される中間文は生成できない。

LLM 手法　LLM に対する few-shot プロンプティングを用いて、所与の文対の中間文候補を生成する。LLM に非逸脱性の条件を満たす中間文のみを生成させることは難しいが、いかなる中間文も生成できる可能性がある。検索拡張生成 [16] の枠組みに従い、事前に収集した (X_s, X_{inter}, X_t) の組から、所与の文対に類似した事例を検索し、これらを few-shot 事例として用いて中間文候補を生成する。

両手法を併用することで、幅広い中間文候補の生成と、それによる分解性能向上が期待できる。

3.1.2　中間文フィルタ

中間文候補を生成する編集操作の文法性と非逸脱性を 2 つの方法で判定し、これらを満たす中間文を選別する。まず、中間文候補 X_{inter} と X_s または X_t との類似度はそれぞれ X_s と X_t 間の類似度よりも高いという非逸脱性の必要条件を、テキスト類似度指標 $sim(\cdot,\cdot)$ を用いて以下のように判定する。

$$sim(X_s, X_{inter}) > sim(X_s, X_t) \text{ かつ}$$
$$sim(X_{inter}, X_t) > sim(X_s, X_t)$$

さらに、所与の (X_s, X_{inter}, X_t) に対して、$X_{inter} = E(X_s)$ である E の文法性と非逸脱性の程度 p_{gu} を算出する編集操作スコアラを用いて、p_{gu} が閾値を超えているかを判定する。編集操作スコアラは、事前学習済み言語モデルを微調整することで実装できる。

3.2　パス探索

中間文のラティスを生成後、以下の 3 ステップで (X_{src}, X_{edt}) をつなぐ編集操作の系列を得る。

1. プリミティブ性の算出　各編集操作、つまりラティスの各エッジのプリミティブらしさを算出する。具体的には、プリミティブ性スコアラを用いて、所与の (X_s, X_{inter}, X_t) に対して、$X_{inter} = E(X_s)$ である E のプリミティブ性の程度 p_{pri} を算出する。プリミティブ性スコアラは、編集操作スコアラと同様の手法で実装できる。

2. 系列の選択　対数尤度 $(\sum \log p_{pri})$ が最大となる文の系列をビタビアルゴリズムを用いて選択する。

3. 編集操作への変換　選択された文の系列 $(X_{src}, \ldots, X_{edt})$ を編集操作抽出器を用いて編集操作の系列 $(E_1, \ldots, E_n)$ に変換する。E_i がプリミティブと仮定し、X_{i-1} と X_i $(= E_i(X_{i-1}))$ の差分を単語マッチングで抽出することで、その差分を E_i とする。

4　実験

提案手法を実装し、その有効性を検証した[1)]。

1)　なお、実装の詳細と評価データの詳細については、それぞれ付録 A と付録 B を参照されたい。

表 1　TS データセットと PE データセットに対する各システム (A: アライメント手法、L: LLM 手法) の結果

	TS						PE					
システム	完全一致率		再現率		適合率		完全一致率		再現率		適合率	
A+L	**0.44**	(22/50)	**0.61**	(90/146)	**0.66**	(90/137)	**0.64**	(32/50)	**0.59**	(51/86)	0.71	(51/72)
A	0.42	(21/50)	0.56	(82/146)	0.65	(82/126)	**0.64**	(32/50)	0.57	(49/86)	**0.72**	(49/68)
L	0.28	(14/50)	0.28	(41/146)	0.62	(41/66)	0.48	(24/50)	0.31	(27/86)	0.60	(27/45)

4.1　実装方法

中間文候補生成器　アライメント手法のフレーズアライナーとして、Enju[2] に基づいてアライメント集合を出力する手法 [17][3] の学習済みモデル[4] を用いた。LLM 手法には、Llama-2 70B [18] を 4bit 量子化したモデルである Llama-2.cpp 70B[5] を使用した。k-shot プロンプティングに使用する k 近傍の組をユークリッド距離に基づいて検索するために、FAISS [19][6] を用いた。k は 5 とし、各文対に対して中間文候補 1 文を生成した。

中間文フィルタ　テキスト類似度指標には、$(1 - TER)$ [20][7] を用いた[8]。編集操作スコアラは RoBERTa [21][9] に全結合層を 1 層追加し、テキスト平易化事例に文法性と非逸脱性を付与したデータセットを用いて微調整することで実現した。編集操作スコアラの閾値は 0.5 とした。

プリミティブ性スコアラ　編集操作スコアラと同様に、テキスト平易化事例にプリミティブ性を付与したデータセットを用いて、RoBERTa と新たな層を微調整することで実現した。

編集操作抽出器　TER で獲得できる単語マッチングを用いて、X_{i-1} を X_i に変換する編集操作を抽出した。編集操作は、X_{i-1} への適用箇所が一意に定まる最短スパンとした。

4.2　評価方法

英語のテキスト平易化 (TS) と機械翻訳後編集 (PE) データセットを用いて評価した。両データセットとも、TER が 0.50 以下の文対を 0.1 刻みに 10 文対ずつ、合計で 50 文対抽出し、各文対に可能な全ての順序の参照系列を付与し、作成した。

システムの評価指標には編集操作系列の完全一致率、編集操作の再現率、適合率を用いた。完全一致率については、システムの出力した編集操作系列が参照系列の 1 つと完全一致したかどうかで評価した。再現率と適合率は、出力したそれぞれの編集操作が参照系列に存在するかどうかで評価した[10]。

4.3　結果

アライメント手法 (A)、LLM 手法 (L)、両手法 (A+L) を用いた 3 つのシステムの比較を行った。

表 1 にそれぞれのデータセットにおける各システムの評価結果を示す。どちらのデータセットにおいても、A+L システムが最も良い評価であり、TS データセットと PE データセットのそれぞれ 44%と 64%の事例を適切な編集操作系列に分解できた。A システムは A+L システムの次に良い評価となった。L システムの再現率が非常に低いのは、LLM 手法では入力された文対に対して中間文候補を 1 文のみ生成するためである。両データセットの結果を比較すると、スコアラの訓練には TS データを用いたにも関わらず、PE データセットの方が高い精度を達成した。これは、PE データセットの方が TS データセットよりも編集操作数の少ない事例が多かったためであるが、提案手法は TS 以外の編集タスクの事例であってもある程度分解できることが示された。

5　おわりに

本稿では、テキスト編集事例の分解タスクを形式化し、アライメントツールと事前学習済み大規模言語モデルを用いて、自動化を試みた。テキスト平易化と機械翻訳後編集のデータセットを用いた実験の結果、それぞれの事例の 44%と 64%を適切な粒度と順序の編集操作系列に分解できた。提案手法の分解性能は、全自動で使用するには十分でないが、人手分析の支援に資する水準にあると考える。

2)　https://github.com/mynlp/enju
3)　https://github.com/yukiar/phrase_alignment_cted
4)　https://zenodo.org/record/4686663
5)　https://huggingface.co/TheBloke/Llama-2-70B-GGUF
6)　https://github.com/facebookresearch/faiss
7)　https://github.com/mjpost/sacrebleu/blob/master/sacrebleu/metrics/ter.py
8)　TER の計算時には shift 操作を含めた。
9)　https://huggingface.co/roberta-large
10)　部分一致する参照系列は複数存在しうるが、評価には出力の編集操作との重なりが最大となるものを使用した。

謝辞

本研究は JSPS 科研費（課題番号：19H05660、23H03689）および KDDI 財団調査研究助成（課題名：平易な文化財情報を執筆・翻訳する技術）の支援を受けた。Newsela からニュース記事のコーパスを提供いただいた。

参考文献

[1] Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. Sentence compression by deletion with LSTMs. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 360–368, 2015.

[2] Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. Exploring neural text simplification models. In **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL) Short Papers**, pp. 85–91, 2017.

[3] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. Approaching neural grammatical error correction as a low-resource machine translation task. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)**, pp. 595–606, 2018.

[4] Gonçalo M. Correia and André F. T. Martins. A simple and effective approach to automatic post-editing with transfer learning. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 3050–3056, 2019.

[5] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 311–318, 2002.

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

[7] Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 7881–7892, 2020.

[8] Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. A survey of evaluation metrics used for NLG systems. **ACM Computing Surveys**, Vol. 55, No. 2, 2022.

[9] Daichi Yamaguchi, Rei Miyata, Sayuka Shimada, and Satoshi Sato. Gauging the gap between human and machine text simplification through analytical evaluation of simplification strategies and errors. In **Findings of the Association for Computational Linguistics: EACL 2023**, pp. 359–375, 2023.

[10] Rei Miyata and Atsushi Fujita. Understanding pre-editing for black-box neural machine translation. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)**, pp. 1539–1550, 2021.

[11] David Heineman, Yao Dou, Mounica Maddela, and Wei Xu. Dancing between success and failure: Edit-level simplification evaluation using SALSA. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 3466–3495, 2023.

[12] Marta Vila, Maria Antònia Martí, and Horacio Rodríguez. Is this a paraphrase? What kind? Paraphrase boundaries and typology. **Open Journal of Modern Linguistics (OJML)**, Vol. 4, pp. 205–218, 2014.

[13] Venelin Kovatchev, M. Antònia Martí, and Maria Salamó. ETPC - A paraphrase identification corpus annotated with extended paraphrase typology and negation. In **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)**, pp. 1384–1392, 2018.

[14] Mayuka Yamamoto and Masaru Yamada. Translation strategies for English-to-Japanese translation. In Rei Miyata, Masaru Yamada, and Kyo Kageura, editors, **Metalanguages for Dissecting Translation Processes: Theoretical Development and Practical Applications**, pp. 80–91. Routledge, London, 2022.

[15] Rémi Cardon and Adrien Bibal. On operations in automatic text simplification. In **Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability (TSAR)**, pp. 116–130, 2023.

[16] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In **Proceedings of the 34th International Conference on Neural Information Processing Systems**, 2020.

[17] Yuki Arase and Jun'ichi Tsujii. Compositional phrase alignment and beyond. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1611–1623, 2020.

[18] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. **CoRR**, Vol. abs/2307.09288, 2023.

[19] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. **IEEE Transactions on Big Data**, Vol. 7, No. 3, pp. 535–547, 2021.

[20] Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In **Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers (AMTA)**, pp. 223–231, 2006.

[21] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. **CoRR**, Vol. abs/1907.11692, 2019.

[22] Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. Neural CRF model for sentence alignment in text simplification. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 7943–7960, 2020.

[23] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In **8th International Conference on Learning Representations (ICLR)**, 2020.

A　実装の詳細

A.1　中間文候補生成器

アライメント手法　フレーズアライナーの閾値は 0.5 と 0.8 に設定し、それぞれのアライメントを獲得した。そして、それぞれの集合の論理和を最終的なアライメントとして使用した。

LLM 手法　few-shot プロンプティングに使用する組 (X_s, X_{inter}, X_t) のデータベースを作成するために、Newsela-Auto [22] を利用した。Newsela-Auto は人手で平易化されたデータのアライメントを自動で取ることで作成されたパラレルコーパスである。Newsela-Auto には、それぞれの元文書に対して、4 段階の平易度の文書が存在する。まず、Newsela-Auto から、それぞれが 1 文で構成され、かつ異なる文である 3 つの平易度の組を抽出した。次に、これらの組が以下の 3 つの基準の 1 つでも満たさない場合、X_{inter} は非逸脱性の条件を満たさないと判断し、その組は除去した。

- $BERTScore(X_s, X_{inter}) > BERTScore(X_s, X_t)$ [23]
- $1 - TER(X_s, X_{inter}) \geq 1 - TER(X_s, X_t) \geq 0.50$
- $diff(X_s, X_{inter}) \subseteq diff(X_s, X_t)$ かつ $diff(X_t, X_{inter}) \subseteq diff(X_t, X_s)$

なお、$diff(\cdot, \cdot)$ は第 1 引数のトークンのうち、第 2 引数に存在しないトークンを返す関数である。3 つ目の条件は、具体的には、“A ⇒ B” を適用してから “B ⇒ A” を適用するような編集操作を取り除く。これらの基準でフィルタリングを行った結果、8,979 の組からなるデータベースを構築できた。

原文側と編集文側の文の対をベクトル化するため、2 文をセパレータトークンで結合し、LLaMa-2.cpp 70B を用いて埋め込んだ。そして、最後のトークンの埋め込みを文対のベクトルとして使用した。

図 3 にプロンプトの例を示す。LLM 手法では 1 文対に対して、中間文候補を 1 文のみ生成し、LLM の出力のうち、プロンプトの終わりから改行コードまでを中間文候補とした。原文と編集文の対を入力する前にシード値 42 で毎回、初期化した。

A.2　編集操作スコアラ

自動データセットと人手データセットの 2 つのデータセットを用いてモデルを訓練した。自動データセットは、LLM 手法で使用した 8,979 の組を正例とし、除去された組から同数の負例を抽出することで作成した。人手データセットを作成するために、まず Newsela-Auto 内にある文対に対して、アライメント手法を用いて中間文候補を自動的に作成した。そして、人手でそれぞれの編集操作の文法性と非逸脱性を判定した。その結果、正例 1,077 事例、負例 3,383 事例の合計 4,460 事例を得た。はじめに、自動データセットを用いてモデルを訓練し、最終層を初期化してから人手データセットで訓練した[11]。

```
SRC: Do you love Large Language Model?
TGT: Do you like LLM?
Rewrite SRC step by step until the rewritten sentence matches TGT.
Do you love Large Language Model?
Do you love LLM?
Do you like LLM?
…
SRC: Long live Large Language Model!
TGT: Hail to LLM
Rewrite SRC step by step until the rewritten sentence matches TGT.
Long live Large Language Model!
```

図 3　LLM 手法に利用したプロンプト。初めの k 文対 (赤) は few-shot 事例、最後の文対 (黒) が中間文候補を生成する対象の文対である。

A.3　プリミティブ性スコアラ

訓練データセットは、編集操作スコアラの人手データセットの正例のプリミティブ性を人手で判定することで作成した。その結果、正例 589 事例、負例 488 事例の合計 1,077 事例を得た。

システムの出力した編集操作系列に p_{pri} が閾値よりも低い編集操作が含まれていた場合、その編集操作を除外して、評価した。プリミティブ性スコアラの閾値は 0.5 とした。

B　評価データの詳細

TS データセットは Newsela-Auto コーパスのテストセットを用いて作成した。

PE データセットは地方自治体の文書を機械翻訳してから、後編集したデータを用いた。原文書は日本語で記述されており、TexTra[12] を用いて英語に機械翻訳した。そして、英語ネイティブであるプロの翻訳家が後編集を行った。

11)　人手データセットのみ、人手データ＋自動データセット＋初期化あり/なしの 3 つのモデルを評価し、最も性能の良いモデルを使用した。

12)　https://mt-auto-minhon-mlt.ucri.jgn-x.jp/