Vol.2025-NL-263 No.24
2025/3/9

品質推定と大規模言語モデルによる機械翻訳対象の自動編集

惟高 日向[1,a)]　藤田 篤[2,b)]　梶原 智之[1,c)]

概要：機械翻訳の品質を向上させる手法の一つに，翻訳対象の文を事前に書き換えることが考えられる．前編集と呼ばれるこの手法は，翻訳が難しい表現を翻訳しやすい（と考えられる）表現に変換することで，訳出時の誤りを低減しようとするものである．これまでに，人手による前編集の有効性の考察や自動化の試みがなされてきた．しかし，前編集の自動化に関する先行研究は，翻訳対象である起点言語文のみを参照するため，対象とする機械翻訳器が適切に翻訳できる（すなわち編集の必要がない）表現を編集することなどによって翻訳品質を低下させてしまう恐れがある．本研究では，前編集ではなく，対象とする機械翻訳器によって得た翻訳文を参照し，実際に生じている誤りを回避するように起点言語文を編集する手法を提案する．具体的には，語単位の品質推定に基づいて翻訳文中の誤りおよび起点言語文中の対応箇所を特定し，当該箇所を大規模言語モデルを用いて自動的に異なる表現に編集する．ただし，編集すべき箇所を特定できたとしても，どのように編集すれば翻訳品質を向上させられるかを事前に知ることは困難である．そこで，編集後の起点言語文を改めて機械翻訳し文単位の品質推定に基づいてより高品質な訳文を選択する，という処理を繰り返すことにより，探索的かつ漸進的に翻訳品質を改善する．日英・日中・英日翻訳に関する実験の結果，翻訳時に誤りを生じていた翻訳対象文に対して，翻訳文を参照しない既存の手法よりも提案手法の方が大きく翻訳品質を改善できることが確認できた．

1. はじめに

所与の機械翻訳器をより効果的に活用する手法の一つとして，前編集と呼ばれるものがある．これは，翻訳対象である起点言語文を事前に翻訳しやすいように書き換える手法である．宮田ら [1,2] は，機械翻訳結果を使いつつ人間が起点言語文を探索的に書き換えることにより，機械翻訳の品質の限界を明らかにした．ここでは，機械翻訳の前編集に有用であるとして，言い換えを含む様々な種類の編集操作が体系化されている．一方，機械翻訳のための前編集の自動化に関する研究 [3–15] では，翻訳対象である起点言語文のみを参照して編集を行っている．対象とする機械翻訳器が適切に翻訳できる（すなわち編集の必要がない）表現を編集するなど，翻訳品質を低下させてしまう恐れがある．編集の種類に関しても，語彙や構造の平易化や，語単位あるいは文単位の言い換えに限定されている．

本研究では，所与の機械翻訳器による翻訳文を参照し，実際に生じている誤りを回避するように起点言語文を編集する手法を提案する．具体的には，まず，翻訳文中の誤りおよびそれらに対応する起点言語文中の表現を，語単位の品質推定によって特定する．そして，誤りに対応する起点言語文中の箇所を，大規模言語モデル (LLM) を用いて自動的に編集する．ただし，どのような表現への編集が翻訳品質の向上に寄与するかを事前に知ることは困難である．そこで，文単位の品質推定によって翻訳品質を逐次評価し，より高品質な翻訳文を選択する．このような処理を繰り返すことにより，探索的かつ漸進的に翻訳品質を改善する．

2. 先行研究

宮田ら [1,2] は，機械翻訳のサービスが社会に浸透してきたことをふまえ，そのようなブラックボックスの機械翻訳器を活用する手段の一つとして起点言語文の編集に着目した．彼らは，統計的機械翻訳 (SMT) およびニューラル機械翻訳 (NMT) を対象とし，起点言語文および翻訳文を参照しながら人手で最小単位の編集を繰り返す作業を実施し，起点言語文の編集による翻訳品質への影響の多寡および有用な編集操作の全体像を明らかにした．

これまでに，機械翻訳前編集の自動化に関する研究も行われてきた．既存の研究は，特定の言語現象を扱うための手法と，翻訳対象文全体を別の文に変換する手法に大別できる．前者は，一般に翻訳が難しいと考えられる言語現象を回避するためのものであり，例えば低頻度語や難語の置

1 愛媛大学大学院理工学研究科
2 情報通信研究機構
a) koretaka@ai.cs.ehime-u.ac.jp
b) atushi.fujita@nict.go.jp
c) kajiwara@cs.ehime-u.ac.jp

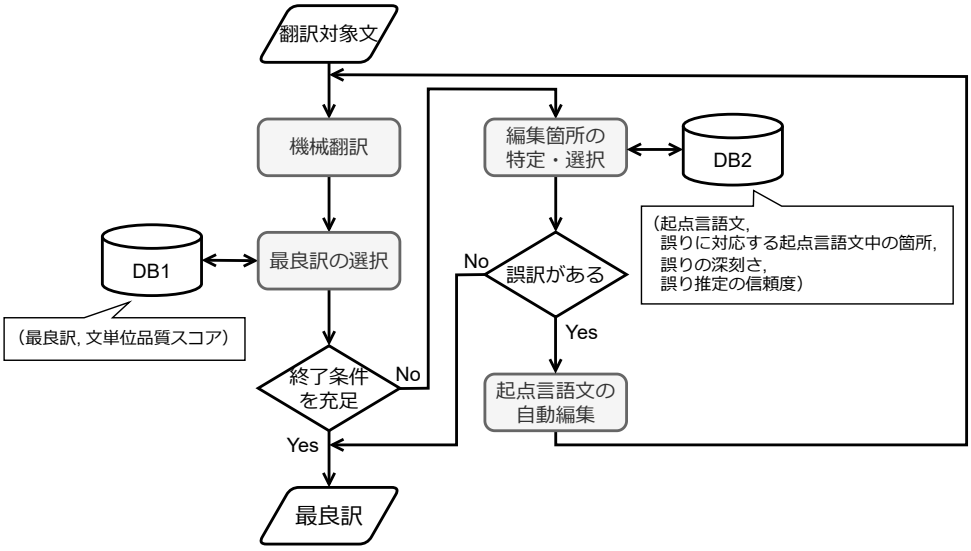

図 1 提案手法の概要

換，省略されている主語の補完，長文の分割などが挙げられる．これらの編集操作を，形態素解析や構文解析などの言語解析器，書き換え規則，コーパスの統計量（単語埋め込みを含む），エンコーダ型の言語モデルなどを用いて実現することで，翻訳品質を向上できることが示されている [3–8, 11–13, 15]．しかしながら，これらの研究で扱われている現象は，宮田らが明らかにした前編集操作の多様性に比して限定的である．また，規則に基づく手法など，他の起点言語への転用が容易ではないものもある．

一方，翻訳対象文全体を別の文に変換する手法は，対訳データに基づく機械翻訳技術（SMT あるいは NMT）を援用し，単一言語内の翻訳として前編集を実現するものである [9–11, 14, 15]．この手法による前編集の能力は，学習に用いられる起点言語の編集前後の文の対のデータ（単言語パラレルデータ）の性質および規模に依存する．このようなデータとして，テキスト平易化のデータ，文書の編集履歴のデータなどを用いることが考えられるが，翻訳の前編集として有用であるとは限らず，規模も限られている．そのため，機械翻訳による逆翻訳や折り返し翻訳によって生成できる擬似的な単言語パラレルデータがよく用いられている．この手法は，規則に基づく機械翻訳や SMT に対しては有効であったと報告されている [9–11] が，NMT に対しては必ずしも翻訳品質の向上に寄与しない [15]．また，この手法は翻訳対象文全体を書き換えるため，特定の言語現象を扱う手法のような編集対象の制御が困難である．

上記の 2 種類の手法のいずれも，宮田らの試みとは異なり，前編集の際に参照するのは翻訳対象である起点言語文のみである．対象とする機械翻訳器が所与の文をどのように翻訳できるかを考慮しないため，適切に翻訳できる（すなわち編集の必要がない）表現を編集するなどして，翻訳品質を低下させてしまう恐れがある．

3. 提案手法

本研究では，翻訳文を参照しつつ起点言語文を自動的に編集することによって，所与の機械翻訳器の翻訳品質を向上させることを試みる．提案手法の概要を図 1 に示す．提案手法ではまず，与えられた起点言語文（以下，**翻訳対象文**）を，対象とする機械翻訳器を用いて翻訳し，得られた翻訳文を最良の訳とする．次に，これらの起点言語文および翻訳文を参照して，起点言語文中の編集すべき箇所を特定する（図 1 の「編集箇所の特定・選択」，3.1 節）．一般に，翻訳文中には複数の誤りが存在し，起点言語文中の対応箇所も複数存在する．我々は，これらを同時に編集することは困難であると考え，最も深刻な誤りに対応する箇所から順に一箇所ずつ編集することによって漸次的に翻訳品質を向上させる．起点言語文中の編集すべき箇所を自動的に異なる表現に編集する（図 1 の「起点言語文の自動編集」，3.2 節）際，どのように編集すれば翻訳品質を向上させられるかを事前に予測することは困難である．そこで，編集後の起点言語文を改めて当該機械翻訳器で翻訳し，それまでに得た訳文の中から最良の翻訳文を選択する（図 1 の「最良訳の選択」，3.3 節）．この一連の処理を，事前に指定した回数あるいは編集すべき箇所がなくなるまで繰り返し実行する（3.4 節）ことによって，探索的かつ漸進的に翻訳品質を改善する．

3.1 編集箇所の特定・選択

所与の起点言語文および翻訳文の対に対して，起点言語文中の編集すべき箇所を特定する．まず，語単位の品質推定器を用いて翻訳文中の（一般に複数の）誤りを検出する．次に，起点言語文と翻訳文の間で単語アライメントを行い，翻訳文中の個々の誤り箇所に対応する起点言語文中の箇所

を特定する．このようにして得られた起点言語文中の複数箇所のうち最も深刻な誤りに対応するものを，編集すべき箇所として選択する．

ここでは，より高品質な訳文の探索（3.4 節で後述）のために，起点言語文中の他の誤り対応箇所の情報を，当該起点言語文，対応する翻訳文中の誤り箇所，当該誤りの深刻さ，品質推定時の信頼度の情報とともに保持しておく（図 1 の DB2）．また，探索の過程では，これらの情報を必要に応じて読み込み，編集すべき箇所を特定する対象とする．

3.2 起点言語文の自動編集

所与の起点言語文中の指定された箇所を，自動的に異なる表現に編集する．編集箇所として多様な表現や構造などが指定されることを想定する必要がある．また，指定された表現の内容および文脈に照らして適切な，多様な編集方法を容易に実現できることが望ましい．そこで我々は，多様な表現を学習済であり，多様な編集処理を実現できると期待できる大規模言語モデル (LLM) を用いることにした．まず，事前に作成した自動編集用プロンプトのテンプレートに起点言語文および当該文中の編集箇所の情報を埋め込み，実際に使用するプロンプトを得る．そして，作成したプロンプトを LLM に入力し，起点言語文の編集文を得る．

ここでは，LLM が異なる表現への編集というタスクを正確に実行するとともに所望の形式を出力するように，プロンプトによってタスクの説明およびタスクの実施例を適切に指示することが重要である．また，入力文がそのまま出力される場合や出力形式に不備がある場合など，LLM の挙動への対処も必要である．

3.3 最良訳の選択

語単位の品質推定器に基づいて特定した編集箇所が実際には誤りに対応していない場合や，誤りに対応していたとしても翻訳品質が向上するように編集できない場合などがある．そこで，編集後の起点言語文を改めて機械翻訳した後，翻訳文の品質を評価して，最良の翻訳文を選択する．個々の翻訳文の品質スコアは，最初に与えられた（未編集の）翻訳対象文と当該翻訳文に対する文単位の品質推定器によって評価する．そして，品質スコアが最も高い訳文を最良の訳文として保持しておく．

3.4 より高品質な訳文の探索

人手による前編集の研究 [1] において，編集を繰り返しても翻訳品質が単調に改善するとは限らず，一方で，編集過程で得た起点言語文から別の編集を試すことや異なる箇所を編集することなどによって，より高品質な訳文が得られる場合があると報告されている．これにならい提案手法でも，深さ優先で最良訳を探索する．すなわち，語単位の品質推定器の出力を手がかりとして貪欲に編集を繰り返すが，編集後の起点言語文に対する訳文の品質が向上しなかった場合は当該編集前の起点言語文における次に深刻な誤りに対応する箇所を編集することとし，訳文中に（編集を未試行の）誤りがなくなった場合は，編集過程で直前に得た起点言語文にバックトラックし，同様に次点の誤りに対応する箇所を選択する．

図 1 では明示していないが，編集箇所の特定・選択の処理（3.1 節）において，検出済の誤りに対応する箇所や当該誤りの深刻さなどの情報を保持しておくことで，探索の機能を実現できる．探索範囲のすべての起点言語文について対応する翻訳文中の誤りおよび起点言語文中の対応箇所が存在しない場合は，編集を行う動機がないため，その時点の最良訳を出力して処理を終了する（図 1 の「誤りがある」に対する “No” の経路）．終了条件としては例えば，探索過程における編集回数に上限を設けることが考えられる．

3.5 事前実験

日英方向の特定の機械翻訳器および当該翻訳器が翻訳誤りを含む起点言語文を対象として，提案手法の洗練および LLM 向けの自動編集用プロンプトの設計・調整を行った．

機械翻訳器としては，JParaCrawl[*1] [16] に基づいて訓練された big モデルを用いた．また，(a) 科学技術論文の抄録から構築された ASPEC[*2] [17] の検証用データ，(b) ニュースや会話など多様なドメインから構築された WMT22[*3] [18] の評価用データ, (c) Reddit から構築された MTNT[*4] [19] の評価用データ, および (d) 京都関連の Wikipedia 記事に対する日英対訳コーパス[*5]のうち KFTT[*6]が定める検証用データを用いた．まず，これら 4 つのデータセットの起点言語文を機械翻訳し，語単位の品質推定器である XCOMET[*7] [20] を用いて誤りを検出した．そして，“critical” または “major” の誤りを含む文を各データセットから 25 文（合計 100 文），ランダムに抽出した．提案手法の設計および洗練は，この 100 文のみを用いて行った．翻訳文中の各誤り箇所に対応する起点言語文中の箇所は，OTAlign[*8] [21] を用いて特定した．その際，単語埋め込みには InfoXLM[*9] [22] を用い，コスト関数をコサイン距離，質量を一様分布，正規化項を 0.1 とした．自動編集には，起点言語である日本語のデータを中心に追加訓練された LLM である Llama-Swallow[*10]を

[*1] https://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/
[*2] https://jipsti.jst.go.jp/aspec/
[*3] https://github.com/wmt-conference/wmt22-news-systems/archive/refs/tags/v1.1.tar.gz
[*4] https://github.com/pmichel31415/mtnt/releases/download/v1.1/MTNT.1.1.tar.gz
[*5] https://alaginrc.nict.go.jp/WikiCorpus/
[*6] https://www.phontron.com/kftt/
[*7] https://huggingface.co/Unbabel/XCOMET-XL
[*8] https://github.com/yukiar/OTAlign
[*9] https://huggingface.co/microsoft/infoxlm-base
[*10] https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-70B-Instruct-v0.1

用いることとし，自動編集用プロンプトにおける起点言語文および編集箇所の指定方法を検討した．

なお，人手による前編集の研究 [2] では，NMT の訳質を十分に向上させるのに平均 5.4〜21.8 回の編集を要したと報告されている．一方，この事前実験を通じて我々は，編集回数 5 回以内で翻訳品質がほぼ上限に達することを確認した．ただしこれは，人手による前編集と比べて品質の改善幅が限定的である（5.2 節）ためであり，各処理の高精度化に応じて，より多くの回数の編集によって翻訳品質をさらに向上させられる可能性がある．

4. 評価実験

提案手法の有用性を確認するために，日英・日中・英日の 3 つの翻訳方向における機械翻訳の品質を評価した．

4.1 実験設定

4.1.1 提案手法の詳細設定

語単位および文単位の品質推定には，XCOMET[*7] [20] を用い，語単位の品質推定では XCOMET が “critical”，“major”，“minor” のいずれかと判断した箇所を誤りとした．翻訳文中の誤りに対応する起点言語文中の表現を特定するための単語アライメントには，OTAlign および INFOXLM を事前実験（3.5 節）と同じ設定で用いた．

提案手法における自動編集は起点言語における単言語処理であるため，起点言語のデータで十分に訓練された LLM が高い性能を発揮すると期待される．そこで，日本語の起点言語文の編集には Llama-Swallow[*10] [23] を，英語の起点言語文の編集には Llama[*11] [24] を用いた．これらのモデルに対する自動編集用プロンプトには，編集というタスクの説明に加えて，起点言語の編集事例 1 例を含めた．日英対照の多様な編集の事例を参考にするため，「言い換えのあれこれ」[*12]に掲載されている言い換え事例（日本語 71 事例および英語 44 事例）を抽出し，事例の候補とした用いた．自動編集用プロンプトでは，対象言語の候補から都度ランダムに選択した 1 例を用いた．実験で使用したプロンプトのテンプレートを付録 A.1 に示す．

提案手法における編集の探索回数の上限（以下，k）は，事前実験（3.5 節）の結果をふまえ 5 回とした．

4.1.2 機械翻訳器

提案手法の適用対象である機械翻訳器として，複数の NMT および LLM を使用した．

NMT としては，TexTra[*13]および NLLB-200[*14] [25] をすべての翻訳方向で使用した．日英翻訳および英日翻訳については，JParaCrawl[*1] [16] に基づいて訓練された 3 種類の公開モデル (small, base, big) も使用した．

LLM としては，目標言語のデータを中心に訓練されたモデルが目標言語方向の翻訳において優れた性能を示す [23] ことが知られているが，検証を兼ねてすべての翻訳方向で Llama[*11]および Llama-Swallow[*10]を用いた．これらのモデルに対する機械翻訳用プロンプトのテンプレートを付録 A.2 に示す．機械翻訳についても事例を 1 例プロンプトに含めた．日英および英日方向の翻訳事例の候補としては WMT23 [26] における各翻訳方向の訓練データ（33,875,119 事例）を用いた．日中の翻訳事例の候補としては JParaCrawl[*1] [27]（4,602,328 事例）を用いた．所与の起点言語文を翻訳する際にプロンプト中で例示する翻訳事例は，上述の候補の中で当該起点言語文に最も類似している起点言語文を BM25 [28] またはベクトルのコサイン類似度を用いて選択し，それに対応する目標言語文とともに用いた．BM25 の実装としては bm25s[*15] [29] を使用し，単語分割手法として日本語は MeCab[*16] [30]，英語は Moses tokenizer[*17] [31] を使用した．ベクトルに基づく事例検索には faiss[*18] [32] を使用し，文ベクトルの取得には多言語文埋め込みモデル LaBSE[*19] [33] を使用した．

4.1.3 評価用データ

日英翻訳の評価には，科学技術論文の抄録から構築された ASPEC [17] の評価用データ，ニュースや会話など多様なドメインから構築された WMT23[*20] [26] の評価用データ，Reddit から構築された MTNT19[*21] [34]，および京都関連の Wikipedia 記事に対する日英対訳コーパス[*5]のうち KFTT[*6]が定める評価用データを用いた．日中翻訳の評価には，ASPEC の評価用データを用いた．英日翻訳の評価には，Wiki ニュースから構築された ALT[*22] [35]，WMT23 の評価用データ，および MTNT19 を用いた．

各評価用データセットにおいて，各機械翻訳器が誤りを生じる文の数を表 1 に示す．以下では，これらを**評価用データの誤り部分集合**と呼ぶ．提案手法は，これらの文に対してのみ自動編集を行い，誤りを含まない文には適用されないことに注意されたい．

4.1.4 評価指標

機械翻訳結果の品質を評価する指標として COMET スコア [36] を使用した．COMET のモデルは，wmt22-comet-

[*11] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct
[*12] https://paraphrasing.org/paraphrase.html
[*13] https://mt-auto-minhon-mlt.ucri.jgn-x.jp/ ver. GPMT-3.9_240731_nmt
[*14] https://huggingface.co/facebook/nllb-200-3.3B
[*15] https://github.com/xhluca/bm25s
[*16] https://taku910.github.io/mecab/
[*17] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl
[*18] https://github.com/facebookresearch/faiss
[*19] https://huggingface.co/sentence-transformers/LaBSE
[*20] https://github.com/wmt-conference/wmt23-news-systems/archive/refs/tags/v.0.1.tar.gz
[*21] https://pmichel31415.github.io/hosting/MTNT2019.tar.gz
[*22] https://huggingface.co/datasets/mutiyama/alt

表 1 各機械翻訳による訳文が誤りを含んでいると判断された文の数

機械翻訳器	日英				日中	英日		
	ASPEC	WMT23	MTNT19	KFTT	ASPEC	ALT	WMT23	MTNT19
総文数	1,812	1,992	1,110	1,160	2,107	1,018	2,074	1,392
TexTra	1,433	1,409	844	1,082	2,037	713	1,601	1,094
NLLB-200	1,233	1,332	766	961	2,026	823	1,638	1,077
JParaCrawl (small)	1,314	1,544	894	1,069	-	900	1,817	1,225
JParaCrawl (base)	1,312	1,514	888	1,065	-	879	1,802	1,214
JParaCrawl (big)	1,224	1,432	862	1,077	-	844	1,755	1,225
Llama (BM25)	1,261	1,321	795	1,037	2,003	767	1,630	1,126
Llama (vector)	1,253	1,331	789	1,040	1,979	748	1,612	1,110
Llama-Swallow (BM25)	1,141	1,262	732	1,029	1,940	729	1,538	1,072
Llama-Swallow (vector)	1,117	1,250	733	1,035	1,914	716	1,525	1,066

da[*23]を使用した[*24]．自動編集を実施しない場合に対する COMET スコアの変化が有意であるか否かを確認するために，有意水準を 5%として，Paired Bootstrap Resampling [38][*25]に基づく統計的仮説検定を行った．

4.1.5 比較手法

評価対象の各機械翻訳器（4.1.2 節）について，翻訳対象文を一切編集しない場合をベースラインとする．また，既存の前編集手法のうち，次の 2 種類の性能を評価した．

比較手法 1: 惟高ら [15] による語レベルの言い換え手法．生成する言い換え文の数は，提案手法の繰り返し回数に揃えて 5 文とした．

比較手法 2: 比較手法 1 のうち，リランキング部分を提案手法で用いる文単位の品質推定器に置き換えた手法．

4.2 評価用データ全体における実験結果

表 2 に評価用データ全体に対する COMET スコアを示す．提案手法は，ほぼすべての機械翻訳器および評価用データの組み合わせに対して，編集なしのベースラインよりも高い COMET スコアを達成した．また，同様に，既存手法（比較手法 1）と同等以上の COMET スコアを示した．しかし，一部の機械翻訳器では，比較手法 1 の方が良い結果を示す場合がある．提案手法は，機械翻訳器として JParaCrawl に基づくモデルを用いた場合，KFTT を除いて比較手法 2 よりも高い COMET スコアを示した．しかし，TexTra，NLLB-200，LLM に基づく機械翻訳器に対しては，比較手法 2 の方が高い COMET スコアを示す場合がある．

4.3 誤りを含む部分集合に対する実験結果

提案手法は，既存の手法と異なり，所与の機械翻訳器が誤りを生じる翻訳対象文のみを編集する．公平な評価のため，評価用データの誤り部分集合（4.1.3 節）に対する COMET スコアを表 3 に示す．提案手法は，評価用データ全体を対象とする場合と同様に，ほとんどの機械翻訳器および評価用データにおいて，編集なしのベースラインよりも高い COMET スコアを達成した．また，提案手法は全 69 種類の機械翻訳器・評価データの組み合わせのうち 46 種類において最も高い COMET スコアを示した．これは，評価用データ全体に対する場合（27 種類）よりも多いことから，提案手法の有用性を示していると言える．ただし，比較手法と比べて改善の幅が小さい場合（13 設定）や前編集による品質の劣化を招いてしまう場合（10 設定）もあった．これらの原因の究明は今後の課題とする．

4.4 機械翻訳器・評価用データごとの傾向

機械翻訳器ごと，評価用データごとの提案手法の効用の傾向を確認するために，評価用データ全体に対する起点言語の翻訳文の COMET スコアと提案手法により得られた最良訳の COMET スコアの上り幅を図 2 のように描画した．また，評価用データの誤り部分集合に対する同様の結果を図 3 に示す．全体的な傾向として，2 つの値の間には負の相関があった．すなわち，編集を行わない場合の翻訳品質が低いほど提案手法による品質の改善幅が大きかった．一般に，誤りをより多く含むほど，また深刻な誤りを含むほど文単位の COMET スコアが低い．すなわち，機械翻訳器が誤りを生じる文を対象として，提案手法によりそのような誤りを解消を試みた結果，誤りを低減し，COMET スコアを改善することができたと考えられる．一方で，編集を行わない場合でも翻訳品質が高い評価用データに対しては COMET スコアの改善が難しいことが確認できた．

なお，翻訳品質が最も改善したのは JParaCrawl (big) を用いた場合の英日 MTNT19，最も劣化したのは TexTra を用いた場合の日英 KFTT であった．

4.5 編集回数に応じた品質の変化

機械翻訳器ごと，評価用データごとに，提案手法の編集

[*23] https://huggingface.co/Unbabel/wmt22-comet-da
[*24] 文献 [37] の報告によると，オープンソースの評価指標の中で COMET スコアは最も人手評価との相関が高い．
[*25] https://github.com/Unbabel/COMET

表 2 評価用データ全体に対する COMET スコア: *は「編集なし」からの有意な変化 ($p < 0.05$),
太字は「編集なし」からの改善，下線は最高値を示す（以下同様）

機械翻訳器	手法	日英				日中	英日		
		ASPEC	WMT23	MTNT19	KFTT	ASPEC	ALT	WMT23	MTNT19
TexTra	編集なし	83.21	80.86	73.74	80.93	88.54	91.43	86.46	82.57
	比較手法 1	**83.58***	80.70	**75.35***	80.86	88.26*	91.33*	86.22*	82.11*
	比較手法 2	**83.52***	**81.05**	**74.79***	80.42*	88.39*	91.39	**86.79***	**82.90***
	提案手法	**83.38***	**81.39***	**75.47***	80.19*	88.19*	91.35	**87.08***	**83.54***
NLLB-200	編集なし	81.21	74.82	68.61	64.30	83.79	88.01	81.43	77.26
	比較手法 1	**81.50***	**75.35***	**68.67**	63.74*	**84.00***	**88.40***	80.93*	76.88
	比較手法 2	**81.50***	**76.43***	**70.52***	**66.93***	**84.08***	**88.73***	**83.38***	**79.47***
	提案手法	**81.71***	**76.32***	**70.28***	**65.82***	**83.89**	**88.57***	**82.68***	**78.45***
JParaCrawl (small)	編集なし	81.78	76.95	72.80	73.49	-	85.77	80.29	74.72
	比較手法 1	**81.97***	**77.22***	**73.18***	**74.04***	-	**85.86**	79.83*	73.21*
	比較手法 2	**81.98***	**77.88***	**73.47***	**73.71**	-	**86.46***	**80.53**	74.07*
	提案手法	**82.28***	**78.57***	**73.99***	**74.16***	-	**86.77***	**82.31***	**77.27***
JParaCrawl (base)	編集なし	82.33	77.78	72.70	74.55	-	86.80	80.99	75.25
	比較手法 1	**82.48***	**78.15***	**73.47***	**75.20***	-	86.74	80.51*	72.74*
	比較手法 2	**82.48***	**78.43***	**73.79***	**74.66**	-	**87.31***	**81.37***	73.79*
	提案手法	**82.76***	**79.19***	**74.32***	**75.19***	-	**87.94***	**83.17***	**77.91***
JParaCrawl (big)	編集なし	82.90	79.29	74.99	76.28	-	88.06	82.33	76.39
	比較手法 1	**83.00**	79.14	74.99	**76.65***	-	87.70*	81.66*	74.45*
	比較手法 2	**82.99**	**79.64***	74.26*	**76.52**	-	**88.32***	**82.80***	75.73*
	提案手法	**83.10***	**80.02***	**75.72***	**76.42**	-	**88.86***	**84.08***	**79.22***
Llama (BM25)	編集なし	81.64	80.48	75.28	76.99	86.60	89.64	85.54	82.34
	比較手法 1	**82.93***	**80.61**	**75.77**	**77.97***	**86.76***	89.56	85.27	81.54*
	比較手法 2	**82.95***	**81.22***	**76.20***	**78.01***	**86.75**	**89.98***	**86.27***	**82.91***
	提案手法	**81.77***	**80.92***	**75.97***	**77.43***	86.58	**89.75**	**86.08***	**83.21***
Llama (vector)	編集なし	82.01	80.70	74.77	76.85	86.40	89.88	85.95	82.47
	比較手法 1	**82.90***	**80.92**	**75.50***	**77.88***	**86.86***	89.57	85.39*	81.44*
	比較手法 2	**82.93***	**81.50***	**76.10***	**77.99***	**86.84***	**89.94**	**86.56***	**83.51***
	提案手法	**82.18***	**81.13***	**75.43***	**76.95**	86.36	**89.93**	**86.38***	**83.05***
Llama-Swallow (BM25)	編集なし	80.87	80.54	75.16	77.18	86.57	90.70	86.43	83.41
	比較手法 1	**82.06***	**81.21***	**76.25***	**78.74***	86.49	**90.92***	86.30	83.14
	比較手法 2	**82.09***	**81.62***	**76.34***	**78.48***	86.40*	**90.98***	**87.14***	**84.11***
	提案手法	80.77	**80.87***	**75.38**	**77.25**	86.19*	**90.94***	**86.97***	**84.11***
Llama-Swallow (vector)	編集なし	81.42	80.95	74.65	77.72	86.51	90.87	86.77	83.49
	比較手法 1	**82.67***	**81.04**	**75.57***	**78.65***	86.39	**91.00**	86.57*	83.19
	比較手法 2	**82.67***	**81.96***	**76.41***	**78.40***	86.39	**91.18***	**87.50***	**84.09***
	提案手法	81.42	**81.32***	**75.34***	77.64	86.08*	**91.03***	**87.20***	**84.32***

回数を変えた場合の COMET スコアおよびその時点までに取りうる最良の翻訳文（参照訳に基づくオラクル訳）の COMET スコアの変化を調査した（付録 A.3）.

例として，翻訳品質が最も改善した設定（JParaCrawl (big) と英日 MTNT19）および最も劣化した設定（TexTra と日英 KFTT）の 2 つを取り上げ，各設定における評価用データの誤り部分集合に対する COMET スコアを図 4 および図 5 に示す．図 4 では，編集を重ねるにつれて単調に翻訳品質が向上していた．これに対して図 5 では，翻訳品質が単調に低下していた．翻訳品質が劣化した他の設定も含めてオラクル訳の COMET スコアは向上していたが，文単位の品質推定器ではオラクル訳を選択することができず，COMET スコアに大きな差が生じていた.

以上より，LLM を用いた起点言語文の自動編集は期待通りに機能していた一方で，文単位の品質推定器に改善の余地があると結論づける.

表 3　評価用データの誤り部分集合に対する COMET スコア

機械翻訳器	手法	日英				日中	英日		
		ASPEC	WMT23	MTNT19	KFTT	ASPEC	ALT	WMT23	MTNT19
TexTra	編集なし	82.43	78.84	71.77	81.17	88.42	90.49	85.04	81.04
	比較手法 1	**82.88***	78.73	**73.74***	81.03	88.13*	90.35*	84.87	80.63*
	比較手法 2	**82.80***	**79.06**	**72.89***	80.63*	88.24*	90.46	**85.47***	**81.34**
	提案手法	**82.64***	**79.60***	**74.05***	80.38*	88.06*	90.37	**85.85***	**82.27***
NLLB-200	編集なし	79.74	72.97	67.94	66.51	84.04	87.51	82.27	78.83
	比較手法 1	**80.13***	**73.28**	67.56	65.54*	**84.16**	**87.66**	81.26*	77.73*
	比較手法 2	**80.10***	**74.31***	**69.36***	**68.22***	**84.19***	**87.97***	**83.25***	**79.67***
	提案手法	**80.48***	**75.22***	**70.36***	**68.34***	**84.14**	**88.19***	**83.85***	**80.37***
JParaCrawl (small)	編集なし	80.41	74.73	71.57	73.49	-	85.01	79.28	73.82
	比較手法 1	**80.65***	**75.15***	**72.01***	**74.00***	-	**85.23**	78.75*	72.23*
	比較手法 2	**80.68***	**75.81***	**72.16***	**73.63**	-	**85.79***	**79.55**	73.08*
	提案手法	**81.11***	**76.82***	**73.05***	**74.20***	-	**86.14***	**81.59***	**76.72***
JParaCrawl (base)	編集なし	81.07	75.40	70.87	74.50	-	85.94	79.89	74.54
	比較手法 1	**81.26***	**75.88***	**71.78***	**75.13***	-	85.94	79.39*	71.88*
	比較手法 2	**81.26***	**76.21***	**72.10***	**74.59**	-	**86.48***	**80.26***	73.01*
	提案手法	**81.66***	**77.25***	**72.90***	**75.20***	-	**87.26***	**82.40***	**77.59***
JParaCrawl (big)	編集なし	81.64	77.12	73.23	76.37	-	87.10	81.02	75.65
	比較手法 1	**81.76**	77.12	**73.31**	**76.72***	-	86.75*	80.33*	73.63*
	比較手法 2	**81.76**	**77.56***	72.20*	**76.60**	-	**87.40***	**81.52***	74.88*
	提案手法	**81.94***	**78.14***	**74.17***	**76.52**	-	**88.06***	**83.09***	**78.86***
Llama (BM25)	編集なし	82.08	79.26	74.26	78.49	86.61	89.04	84.79	81.42
	比較手法 1	**82.42***	79.22	74.25	**78.76**	**86.66**	88.80	84.32*	80.57*
	比較手法 2	**82.44***	**79.72***	**74.54**	**78.88***	**86.65**	**89.27**	**85.35***	**81.84**
	提案手法	**82.27***	**79.93***	**75.21***	**78.98***	86.58	**89.18**	**85.49***	**82.49***
Llama (vector)	編集なし	82.13	79.51	73.78	78.28	86.64	89.25	84.92	81.55
	比較手法 1	**82.36***	79.49	**73.90**	**78.51**	**86.75**	88.83*	84.19*	80.39*
	比較手法 2	**82.36***	**79.95***	**74.47***	**78.56**	**86.72**	**89.26**	**85.51***	**82.39***
	提案手法	**82.37***	**80.16***	**74.69***	**78.40**	86.61	**89.32**	**85.48***	**82.28***
Llama-Swallow (BM25)	編集なし	82.36	79.82	75.31	78.79	86.65	89.84	85.34	82.30
	比較手法 1	82.22	79.61	75.06	**79.15***	86.40*	**90.14***	85.21	81.81*
	比較手法 2	82.15*	**79.92**	74.92	**78.89**	86.32*	**90.22***	**86.09***	**82.77***
	提案手法	82.20	**80.34***	**75.65**	**78.87**	86.24*	**90.17***	**86.06***	**83.22***
Llama-Swallow (vector)	編集なし	82.32	80.05	74.52	78.88	86.62	90.02	85.77	82.16
	比較手法 1	82.28	79.31*	74.00	**79.09**	86.31*	**90.13**	85.50*	81.90
	比較手法 2	82.27	**80.25**	**75.02**	78.81	86.32*	**90.34***	**86.49***	**82.77***
	提案手法	82.30	**80.64***	**75.57***	78.80	86.15*	**90.24***	**86.35***	**83.24***

5. 分析

提案手法の各処理で採用した手法の是非を，2 つの比較実験を通じて分析した．編集箇所の特定手法に関しては，語単位の品質推定器および単語アラインメントに基づく手法の有効性を確認するため，他の手法と比較した．また，自動編集に関しては，訓練データの主たる言語およびサイズの異なる LLM の性能を比較した．

事前実験（3.5 節）で用いた日英方向の誤りを含む 100 文，および JParaCrawl に基づいて訓練されたモデル (big) を用いた．

5.1 編集箇所の特定手法に関する分析

提案手法は，起点言語文における編集すべき箇所を，語単位の品質推定器と単語アライメントを用いて特定する（3.1 節）．これによる翻訳品質の向上は確認できたが，品質推定器に基づいて誤り箇所を特定することが本質的に品質の改善に寄与しているかどうかは明らかではない．

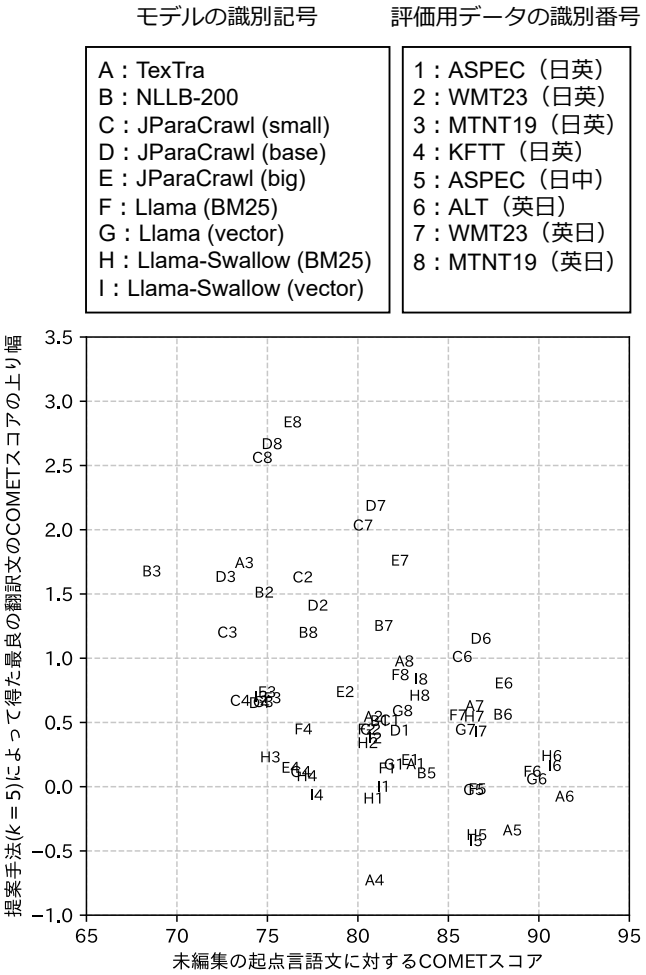

図 2 評価用データ全体に対する COMET スコアと提案手法 ($k = 5$) による COMET スコアの変化 ($r = -0.49$)

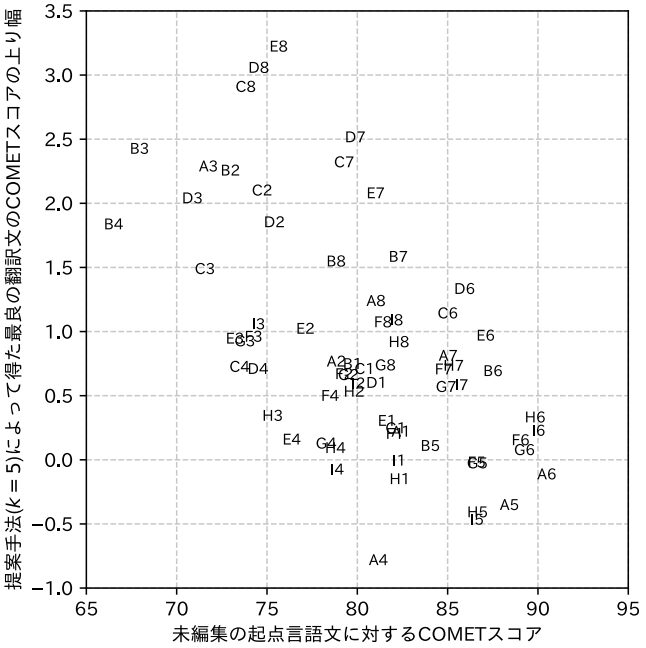

図 3 評価用データの誤り部分集合に対する COMET スコアと提案手法 ($k = 5$) による COMET スコアの変化 ($r = -0.59$)

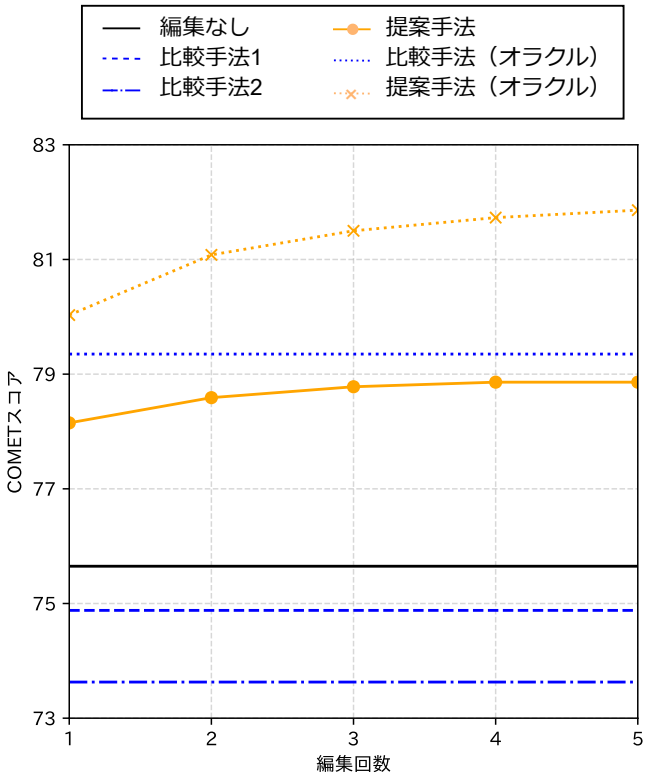

図 4 JParaCrawl (big) および英日 MTNT19 の誤り部分集合に対する COMET スコア: 編集回数ごとの変化

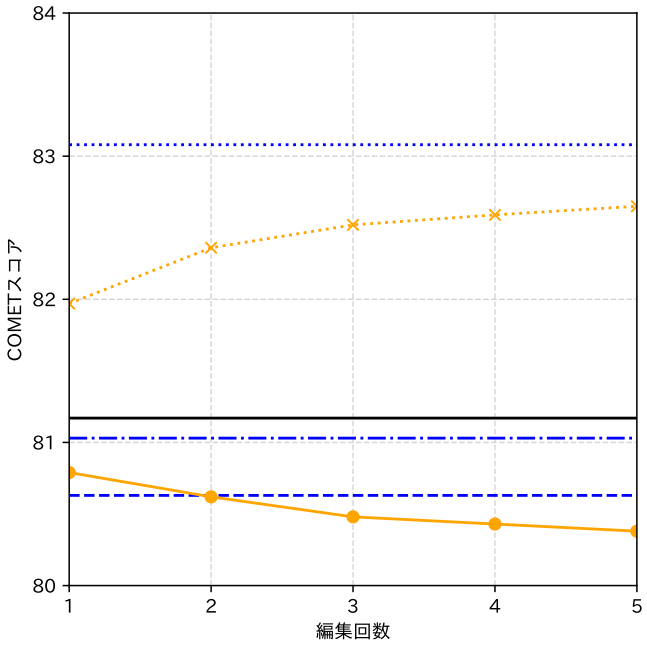

図 5 TexTra および日英 KFTT の誤り部分集合に対する COMET スコア: 編集回数ごとの変化

そこで，品質向上が品質推定器による誤り箇所の特定によるものであるか，編集箇所によらず表現を変更することによるかを明らかにするために，編集箇所を特定する手法として次の 3 種類を比較した．

順入力: 提案手法で採用した，語単位の品質推定と単語アライメントの 2 段階で編集すべき箇所を特定する手

表 4 編集箇所の特定手法を変化させたときの COMET スコア: k は提案手法における編集回数（以下同様）

	順入力	逆入力	ランダム
編集なし	71.82	71.82	71.82
$k=1$	**73.28***	**72.33**	**72.75***
$k=2$	**73.65***	**72.60**	**72.89***
$k=3$	**73.48***	**72.69***	**72.72**
$k=4$	**73.39***	**72.73***	**72.72**
$k=5$	**73.31***	**73.04***	**72.72**
オラクル ($k=5$)	75.42	74.43	73.76

法．誤りが伝搬・蓄積する可能性がある．

逆入力: 語単位の品質推定器である XCOMET への入力の際，翻訳文を起点言語文，起点言語文を翻訳文とみなすことで，起点言語文における編集すべき箇所を直接特定する手法．順入力とは異なり単語アラインメントが不要である．

ランダム: 編集すべき箇所の開始位置を起点言語文においてランダムに決定し，終了位置もその位置から文末までの範囲でランダムに決定する手法．

表 4 に実験結果を示す．編集すべき箇所をランダムに選択した場合でも，翻訳対象文を編集することによって翻訳品質を改善できていた．ただし，品質推定器を用いた 2 つの手法（順入力および逆入力）は，対象をランダムに定めるよりも高い COMET スコアを達成しており，品質推定器を活用することの有効性を示している．特に，提案手法で採用した順入力の手法は，他の手法よりもオラクルの COMET スコアが 1 ポイント程度高く，実際の COMET スコアも編集回数によらず高かった．

順入力の手法では語単位の品質推定と単語アライメントを組み合わせているため，各々の誤りが重畳していると考えられる．したがって，各技術の性能向上によって更に性能を改善できると期待できる．一方，品質推定の分野では，翻訳文と同時に起点言語文における誤りを推定する試みがある [39–41]．近年のシェアードタスクでは翻訳文における誤りのみが扱われているが，翻訳文において訳出されていない（訳抜けしている）起点言語文中の範囲や翻訳文における誤りに対応する起点言語文中の範囲を特定することは，特に起点言語文の編集には重要 [42] であり，データの設計も含めて検討する余地が残されている．

5.2 自動編集に用いるモデルに関する分析

提案手法では，起点言語のデータを中心に訓練された LLM を起点言語文の自動編集（3.2 節）に用いた．しかし，そのようなモデルが最適であるとは限らず，また大規模なモデルの方が一般的に高性能ではあるものの，計算資源および処理速度の面では小規模なモデルの方が望ましい．

そこで，訓練データの主な言語やサイズが異なる LLM を用いた場合の性能を比較した．品質推定器と単語アライメントに基づく手法（5.1 節における順入力の手法）によって編集すべき箇所を特定するなど，自動編集に用いるモデル以外は共通の設定とし，次の 4 種類のモデルを比較した．

LS-70B: Llama-Swallow 70B[*10]，継続事前学習によって起点言語である日本語に適応済み．

LS-8B: Llama-Swallow 8B[*26]，同適応済み．

L-70B: Llama 70B[*11]，同未適応．

L-8B: Llama 8B[*27]，同未適応．

さらに，自動編集の効果の多寡を測るために，提案手法によって自動的に特定した編集すべき箇所を人手で編集する実験も行った．この編集は，第一著者が担当した．

実験の結果，表 5 に示すように，編集回数（k の値）にかかわらず，人手による編集，LS-70B，L-70B，LS-8B，L-8B の順に高い COMET スコアを示した．同じ規模のモデルを比較すると，起点言語に適応済みのモデル (LS-*) の方が未適応のモデル (L-*) よりも翻訳品質を向上させる編集ができることが分かった．起点言語に未適応の L-70B を用いた場合のオラクル訳の品質は，起点言語に適応済の LS-70B と同程度であるが，実際に最良訳を自動的に選択する場合は LS-70B ほどの効果を得ることができなかった．また，起点言語への適応の有無が等しい場合，大規模なモデル (*-70B) の方が小規模なモデル (*-8B) よりも高い COMET スコアを達成できることが分かった．小規模なモデルのうち，起点言語に適応済の LS-8B は，有意ではないものの COMET スコアを改善できた．一方，起点言語に未適応の L-8B を用いた場合，オラクル訳であっても編集なしの場合と同じ COMET スコアであり，自動編集の効果がまったくなかった[*28]．小規模かつ起点言語にも適応していないモデルは指示を理解する能力が低いこと，指示に従って指定した形式の出力をする能力が低いことを示唆している．以上より，LLM の中では編集対象である起点言語に適応済みのモデル，大規模なモデルの方が，翻訳品質をより大きく向上させられることが分かった．

オラクル訳に対する COMET スコアの比較から，現状の LLM およびその使い方では，人手による編集と同様に翻訳品質を高める編集を実現できていないことが分かった．最良訳を自動的に選択した場合とオラクル訳の COMET スコアの差は，最良訳の選択に使用した文単位の品質推定器についてもさらなる改良の余地があることを示唆している．

6. おわりに

機械翻訳のための自動前編集に関する先行研究では，対

[*26] https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.1

[*27] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[*28] 文単位の品質推定に加えて，編集の出力の形式に不備がある場合は編集前の文をそのまま出力するという安全策を講じている．

表 5 自動編集に用いるモデルを変化させたときの COMET スコア

	人手	LS-70B	LS-8B	L-70B	L-8B
編集なし	71.82	71.82	71.82	71.82	71.82
$k = 1$	**73.58***	**73.28***	**72.39**	**72.90***	71.82
$k = 2$	**74.20***	**73.65***	**72.45**	**73.04***	71.82
$k = 3$	**74.50***	**73.48***	**72.67**	**72.95***	71.82
$k = 4$	**74.40***	**73.39***	**72.64**	**73.05***	71.82
$k = 5$	**74.22***	**73.31***	**72.66**	**72.99***	71.82
オラクル ($k = 5$)	77.61	75.42	74.78	75.50	71.82

象とする機械翻訳器による翻訳文を参照することなく起点言語文の編集操作を行っていた．これに対して本稿では，人手による起点言語文編集の分析 [1,2] にならって翻訳文を参照し，実際に生じている誤りを回避するように前編集を行う手法を提案した．具体的には，語単位の品質推定器を用いて誤りを検出する，単語アラインメント技術を用いて起点言語文中の編集すべき箇所を特定する，LLM を用いて起点言語文の所定の箇所を異なる表現に編集する，文単位の品質推定器を用いてより良い翻訳文を選択する，という一連の処理を繰り返し実行することによって，探索的かつ漸進的に翻訳品質を改善することを試みた．

日英・日中・英日翻訳に関する実験の結果，翻訳時に誤りを生じていた翻訳対象文に対して，翻訳文を参照しない既存の手法よりも提案手法の方が大きく翻訳品質を改善できることが確認できた．また，分析を通じて，語単位の品質推定器を用いて編集箇所を特定することが有用であること，自動編集に用いる LLM としては起点言語のデータを中心に訓練された大規模なモデルが有用であること，文単位の品質推定器には改善の余地があることが明らかになった．

謝辞 本研究の成果は，国立研究開発法人情報通信研究機構（NICT）の委託研究（課題番号：22501）により得られたものです．

参考文献

[1] Miyata, R. and Fujita, A.: Dissecting Human Pre-Editing toward Better Use of Off-the-Shelf Machine Translation Systems, *Proceedings of the 20th Annual Conference of the European Association for Machine Translation*, pp. 54–59 (2017).

[2] Miyata, R. and Fujita, A.: Understanding Pre-Editing for Black-Box Neural Machine Translation, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1539–1550 (2021).

[3] Shirai, S., Ikehara, S. and Kawaoka, T.: Effects of Automatic Rewriting of Source Language within a Japanese to English MT System, *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 226–239 (1993).

[4] 金　淵培，江原暉将：日英機械翻訳のための日本語長文自動短文分割と主語の補完，情報処理学会論文誌， Vol. 35, No. 6, pp. 1018–1028 (1994).

[5] 山口昌也，乾　伸雄，小谷善行，西村恕彦：前編集結果を利用した前編集自動化規則の獲得，情報処理学会論文誌， Vol. 39, No. 1, pp. 17–28 (1998).

[6] Shirai, S., Ikehara, S., Yokoo, A. and Ooyama, Y.: Automatic Rewriting Method for Internal Expressions in Japanese to English MT and Its Effects, *Proceedings of the Second International Workshop on Controlled Language Applications*, pp. 62–75 (1998).

[7] 吉見毅彦，佐田いち子：英字新聞記事見出し翻訳の自動前編集による改良，自然言語処理， Vol. 7, No. 2, pp. 27–43 (2000).

[8] 吉見毅彦，佐田いち子，福持陽士：頑健な英日機械翻訳システム実現のための原文自動前編集，自然言語処理，Vol. 7, No. 4, pp. 99–117 (2000).

[9] Sun, Y., O'Brien, S., O'Hagan, M. and Hollowood, F.: A Novel Statistical Pre-Processing Model for Rule-Based Machine Translation System, *Proceedings of the 14th Annual Conference of the European Association for Machine Translation* (2010).

[10] 南條浩輝，山本祐司，吉見毅彦：機械翻訳の品質向上のための対訳コーパスからの統計的前編集システムの自動構築，情報処理学会論文誌， Vol. 53, No. 6, pp. 1644–1653 (2012).

[11] Mirkin, S., Venkatapathy, S., Dymetman, M. and Calapodescu, I.: SORT: An Interactive Source-Rewriting Tool for Improved Translation, *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 85–90 (2013).

[12] Štajner, S. and Popovic, M.: Can Text Simplification Help Machine Translation?, *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pp. 230–242 (2016).

[13] Štajner, S. and Popović, M.: Improving Machine Translation of English Relative Clauses with Automatic Text Simplification, *Proceedings of the First Workshop on Automatic Text Adaptation*, pp. 39–48 (2018).

[14] Mehta, S., Azarnoush, B., Chen, B., Saluja, A., Misra, V., Bihani, B. and Kumar, R.: Simplify-Then-Translate: Automatic Preprocessing for Black-Box Translation, *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8488–8495 (2020).

[15] Koretaka, H., Kajiwara, T., Fujita, A. and Ninomiya, T.: Mitigating Domain Mismatch in Machine Translation via Paraphrasing, *Proceedings of the 10th Workshop on Asian Translation*, pp. 29–40 (2023).

[16] Morishita, M., Chousa, K., Suzuki, J. and Nagata, M.: JParaCrawl v3.0: A Large-scale English-Japanese Parallel Corpus, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 6704–6710 (2022).

[17] Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S. and Isahara, H.: ASPEC: Asian Scientific Paper Excerpt Corpus, *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pp. 2204–2208 (2016).

[18] Kocmi, T., Bawden, R., Bojar, O., Dvorkovich, A., Federmann, C., Fishel, M., Gowda, T., Graham, Y., Grundkiewicz, R., Haddow, B., Knowles, R., Koehn, P., Monz, C., Morishita, M., Nagata, M., Nakazawa, T., Novák, M., Popel, M. and Popović, M.: Findings of the 2022 Conference on Machine Translation (WMT22), *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 1–45 (2022).

[19] Michel, P. and Neubig, G.: MTNT: A Testbed for Machine Translation of Noisy Text, *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing, pp. 543–553 (2018).
[20] Guerreiro, N. M., Rei, R., Stigt, D. v., Coheur, L., Colombo, P. and Martins, A. F. T.: xcomet: Transparent Machine Translation Evaluation through Fine-grained Error Detection, *Transactions of the Association for Computational Linguistics*, Vol. 12, pp. 979–995 (2024).
[21] Arase, Y., Bao, H. and Yokoi, S.: Unbalanced Optimal Transport for Unbalanced Word Alignment, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3966–3986 (2023).
[22] Chi, Z., Dong, L., Wei, F., Yang, N., Singhal, S., Wang, W., Song, X., Mao, X.-L., Huang, H. and Zhou, M.: InfoXLM: An Information-Theoretic Framework for Cross-Lingual Language Model Pre-Training, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3576–3588 (2021).
[23] Fujii, K., Nakamura, T., Loem, M., Iida, H., Ohi, M., Hattori, K., Shota, H., Mizuki, S., Yokota, R. and Okazaki, N.: Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities, *Proceedings of the First Conference on Language Modeling* (2024).
[24] Llama Team: The Llama 3 Herd of Models, *arXiv:2407.21783* (2024).
[25] NLLB Team, Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A., Akula, B., Barrault, L., Mejia-Gonzalez, G., Hansanti, P., Hoffman, J., Jarrett, S., Sadagopan, K. R., Rowe, D., Spruit, S., Tran, C., Andrews, P., Ayan, N. F., Bhosale, S., Edunov, S., Fan, A., Gao, C., Goswami, V., Guzmán, F., Koehn, P., Mourachko, A., Ropers, C., Saleem, S., Schwenk, H. and Wang, J.: No Language Left Behind: Scaling Human-Centered Machine Translation, *arXiv:2207.04672* (2022).
[26] Kocmi, T., Avramidis, E., Bawden, R., Bojar, O., Dvorkovich, A., Federmann, C., Fishel, M., Freitag, M., Gowda, T., Grundkiewicz, R., Haddow, B., Koehn, P., Marie, B., Monz, C., Morishita, M., Murray, K., Nagata, M., Nakazawa, T., Popel, M., Popović, M. and Shmatova, M.: Findings of the 2023 Conference on Machine Translation (WMT23): LLMs Are Here but Not Quite There Yet, *Proceedings of the Eighth Conference on Machine Translation*, pp. 1–42 (2023).
[27] Nagata, M., Morishita, M., Chousa, K. and Yasuda, N.: A Japanese-Chinese Parallel Corpus Using Crowdsourcing for Web Mining, *arXiv:2405.09017* (2024).
[28] Robertson, S. and Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond, *Foundations and Trends in Information Retrieval*, Vol. 3, No. 4, p. 333–389 (2009).
[29] Lù, X. H.: BM25S: Orders of magnitude faster lexical search via eager sparse scoring, *arXiv:2407.03618* (2024).
[30] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 230–237 (2004).
[31] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation, *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pp. 177–180 (2007).
[32] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L. and Jégou, H.: The Faiss library, *arXiv:2401.08281* (2024).
[33] Feng, F., Yang, Y., Cer, D., Arivazhagan, N. and Wang, W.: Language-agnostic BERT Sentence Embedding, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 878–891 (2022).
[34] Li, X., Michel, P., Anastasopoulos, A., Belinkov, Y., Durrani, N., Firat, O., Koehn, P., Neubig, G., Pino, J. and Sajjad, H.: Findings of the First Shared Task on Machine Translation Robustness, *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 91–102 (2019).
[35] Riza, H., Purwoadi, M., Gunarso, Uliniansyah, T., Ti, A. A., Aljunied, S. M., Mai, L. C., Thang, V. T., Thai, N. P., Chea, V., Sun, R., Sam, S., Seng, S., Soe, K. M., Nwet, K. T., Utiyama, M. and Ding, C.: Introduction of the Asian Language Treebank, *Proceedings of the 2016 Conference of the Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Technique (O-COCOSDA)*, pp. 1–6 (2016).
[36] Rei, R., Stewart, C., Farinha, A. C. and Lavie, A.: COMET: A Neural Framework for MT Evaluation, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 2685–2702 (2020).
[37] Freitag, M., Rei, R., Mathur, N., Lo, C.-k., Stewart, C., Avramidis, E., Kocmi, T., Foster, G., Lavie, A. and Martins, A. F. T.: Results of WMT22 Metrics Shared Task: Stop Using BLEU – Neural Metrics Are Better and More Robust, *Proceedings of the Seventh Conference on Machine Translation*, pp. 46–68 (2022).
[38] Koehn, P.: Statistical Significance Tests for Machine Translation Evaluation, *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 388–395 (2004).
[39] Specia, L., Blain, F., Fomicheva, M., Fonseca, E., Chaudhary, V., Guzmán, F. and Martins, A. F. T.: Findings of the WMT 2020 Shared Task on Quality Estimation, *Proceedings of the Fifth Conference on Machine Translation*, pp. 743–764 (2020).
[40] Specia, L., Blain, F., Fomicheva, M., Zerva, C., Li, Z., Chaudhary, V. and Martins, A. F. T.: Findings of the WMT 2021 Shared Task on Quality Estimation, *Proceedings of the Sixth Conference on Machine Translation*, pp. 684–725 (2021).
[41] Fomicheva, M., Lertvittayakumjorn, P., Zhao, W., Eger, S. and Gao, Y.: The Eval4NLP Shared Task on Explainable Quality Estimation: Overview and Results, *Proceedings of the Second Workshop on Evaluation and Comparison of NLP Systems*, pp. 165–178 (2021).
[42] 島田紗裕華，山口大地，宮田 玲，藤田 篤，梶原智之，佐藤理史：機械翻訳向け原文編集の支援に向けた日英翻訳品質推定データセットの設計と構築，言語処理学会第30回年次大会発表論文集，pp. 845–850 (2024).

付 録

A.1 自動編集用プロンプトのテンプレート

日本語文と英語文の自動編集に用いたプロンプトのテン

日本語文の言い換え文を出力してください。
手順は次の通りです。
1. 日本語文の ¶ ¶ で囲まれた言い換え対象表現について、同じ意味を持つ異なる表現の言い換え候補を 1 個から 5 個挙げてください。
2. 手順 1 で挙げた言い換え候補の中から、日本語文の ¶ ¶ で囲まれた箇所を言い換えるのに、最も適切な候補を 1 つ選んでください。
3. 日本語文の ¶ ¶ で囲まれた箇所を、手順 2 で選ばれた最適な言い換え候補を使用して置換してください。この際、以下の基準を満たすように文脈に合わせて適切に調整してください。
・文脈に応じて、適切な動詞の活用形や助詞を使うこと。
・元の文の意味を正確に伝えること。
・文法的に正しい構文を持つこと。

入力例:
日本語文: {{paraphrase[”example_original”]}}
言い換え対象表現: {{paraphrase[”example_original_span”]}}
出力例:
{”言い換え文”:”{{paraphrase[”example_paraphrase”]}}”}

日本語文: {{paraphrase[”annotated_src”]}}
言い換え対象表現: {{paraphrase[”propagate_error_span”]}}

図 A·1 日本語についての自動編集プロンプトのテンプレート

Please output a paraphrased sentence for a given English sentence.
The procedure is as follows:
1. For the target expression for paraphrasing marked with ¶ ¶ in the English sentence, provide 1 to 5 paraphrase candidates with the same meaning and different expressions.
2. Select one among the paraphrase candidates generated in step 1 that is most appropriate for paraphrasing the part marked with ¶ ¶ in the English sentence.
3. Replace the part marked with ¶ ¶ in the English sentence with the paraphrase candidate selected in step 2. At the same time, please perform necessary adjustment to make it fit the context while meeting the following criteria.
・Use appropriate conjugation form of words and particles according to the context.
・Convey the original meaning of the sentence accurately.
・Maintain the grammatically correct structure of the sentence.

Input example:
English sentence: {{paraphrase[”example_original”]}}
Target expression for paraphrasing: {{paraphrase[”example_original_span”]}}
Output example:
{”paraphrased_sentence”:”{{paraphrase[”example_paraphrase”]}}”}

English sentence: {{paraphrase[”annotated_src”]}}
Target expression for paraphrasing: {{paraphrase[”propagate_error_span”]}}

図 A·2 英語についての自動編集プロンプトのテンプレート

プレートを各々図 A·1 および図 A·2 に示す．提案手法において起点言語文を自動的に編集する際，当該起点言語のデータを中心に訓練された LLM を用いた．LLM は一般に，訓練時に使用されたテキストデータの言語ごとの量に応じて，プロンプトに対する理解の程度が異なると考えられる．そこで，日本語のデータを中心に（追加）訓練された Llama-Swallow には日本語のプロンプトを，英語のデータを中心に訓練された Llama には英語のプロンプトを与えた．テンプレート中の “{{” と “}}” で囲まれた箇所はプレースホルダを表し，事例に応じて対応するテキストに置換することにより実際のプロンプトを得る．プレースホルダと対象テキストの対応は表 A·1 に示す通りである．

```
{{source_language}} を {{target_language}} に翻訳してください。

入力例:
英語文: {{translation["example_src"]}}
出力例:
{"翻訳文":"{{translation["example_tgt"]}}"}

英語文: {{translation["src"]}}
```

図 A·3 Llama-Swallow 向けの機械翻訳用プロンプトのテンプレート

```
Please translate the {{source_language}} sentence into {{target_language}}.

Input example:
Japanese sentence: {{translation["example_src"]}}
Output example:
{"translation":"{{translation["example_tgt"]}}"}

{{source_language}} sentence: {{translation["src"]}}
```

図 A·4 Llama 向けの機械翻訳用プロンプトのテンプレート

表 A·1 自動編集用プロンプトのテンプレート中のプレースホルダ

プレースホルダ	対象テキスト
example_original	編集前の文の例
example_original_span	上記において編集された箇所
example_paraphrase	上記に対応する編集後の文
annotated_src	編集対象の文
propagate_error_span	上記において編集すべき箇所

表 A·2 機械翻訳用プロンプトのテンプレート中のプレースホルダ

プレースホルダ	対象テキスト
source_language	起点言語
target_language	目標言語
example_src	起点言語文の例
example_tgt	上記に対応する目標言語文
src	翻訳対象の文

A.2 機械翻訳用プロンプトのテンプレート

Llama-Swallow 向けおよび Llama 向けの機械翻訳用プロンプトのテンプレートを各々図 A·3 および図 A·4 に示す．自動編集用プロンプトのテンプレートと同様に，日本語のデータで（追加）訓練された Llama-Swallow には日本語のプロンプトを，英語のデータで訓練されたモデルには英語のプロンプトを与えた．テンプレート中の “{{” と “}}” で囲まれた箇所はプレースホルダを表し，事例に応じて対応するテキストに置換することにより実際のプロンプトを得る．プレースホルダと対象テキストの対応は表 A·2 に示す通りである．

A.3 編集回数に応じた COMET スコアの変化

図 A·5 に，すべての機械翻訳器およびデータセットの，評価用データ全体に対する提案手法およびオラクル訳の COMET スコアの，編集回数に応じた変化を示す．また図 A·6 に，評価用データの誤り部分集合に対する同様の結果を示す．全体的に，編集回数が多くなるにつれて，提案手法およびオラクル訳の COMET スコアが改善することが確認できた．また，NMT を対象とした場合，評価用データ全体に対しては既存手法の方がオラクル訳の COMET スコアが高いが，評価用データの誤り部分集合に対しては提案手法の方がオラクル訳の COMET スコアが高いことが分かった．このことから，翻訳対象文に対する翻訳品質に応じて本手法と既存手法を使い分けることで，より高品質な翻訳文が得られる可能性がある．一方，LLM に対する結果（下部の 4 段）ではこのような傾向はみられなかった．

評価用データに着目すると，日英・日中 ASPEC（左から 1 列目と左から 5 列目）および日英 KFTT（左から 4 列目）については，評価用データごとの傾向（4.4 節）と同様に，編集を行わない場合の翻訳品質が低い場合には若干の改善が見られるが，翻訳品質が相対的に高い場合には品質が低下する傾向が観察された．オラクル COMET スコアは単調に増加しているが，文単位の品質推定器は一定以上の品質の訳文間の差異を的確に捉えることができていないと考えられる．これらの翻訳用データについては，翻訳文が一定の品質を超えた場合にそれらの品質の差異を捉えられないと考えられる．すなわち，4.5 節で述べたように，文単位品質推定器の課題が示唆される．

A.4 文単位品質スコアと COMET スコアの相関

図 A·7 に，未編集の起点言語文に対する翻訳文と提案手法 ($k = 5$) によって得た最良の翻訳文の文単位品質スコアの差分と COMET スコアの差分を示す．この図から，文単位の品質推定器によって翻訳品質が高い翻訳文を選択したとしても，COMET スコアが改善しない場合があることが分かる．ただし，文単位の品質推定器において翻訳品質が大きく（例えば 25 ポイント以上）改善した場合は，COMET スコアも改善する場合が多い．

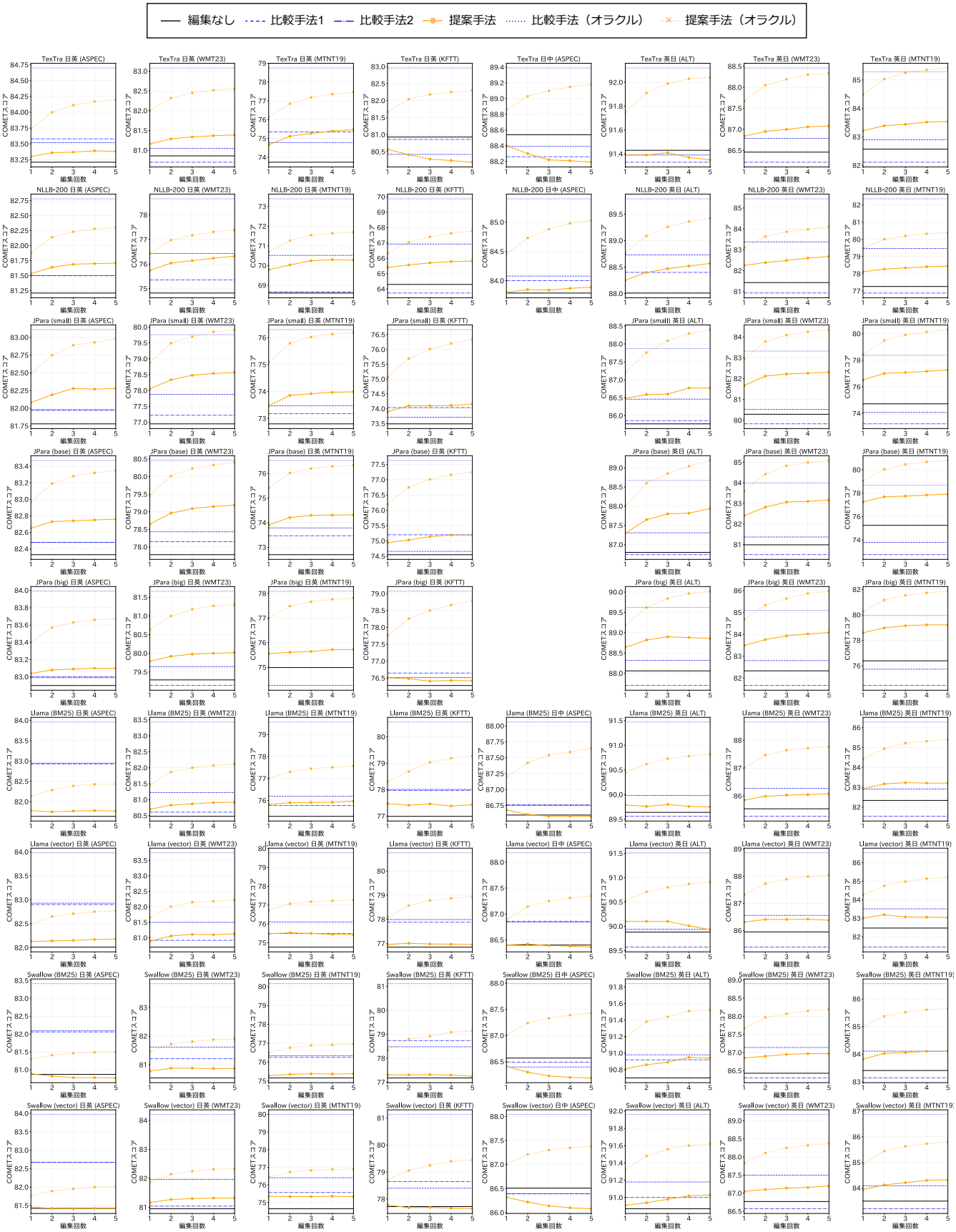

図 A·5 評価用データ全体に対する COMET スコア: 編集回数ごとの変化（図中の “JPara” は JParaCrawl，“Swallow” は Llama-Swallow を指す，以下同様）

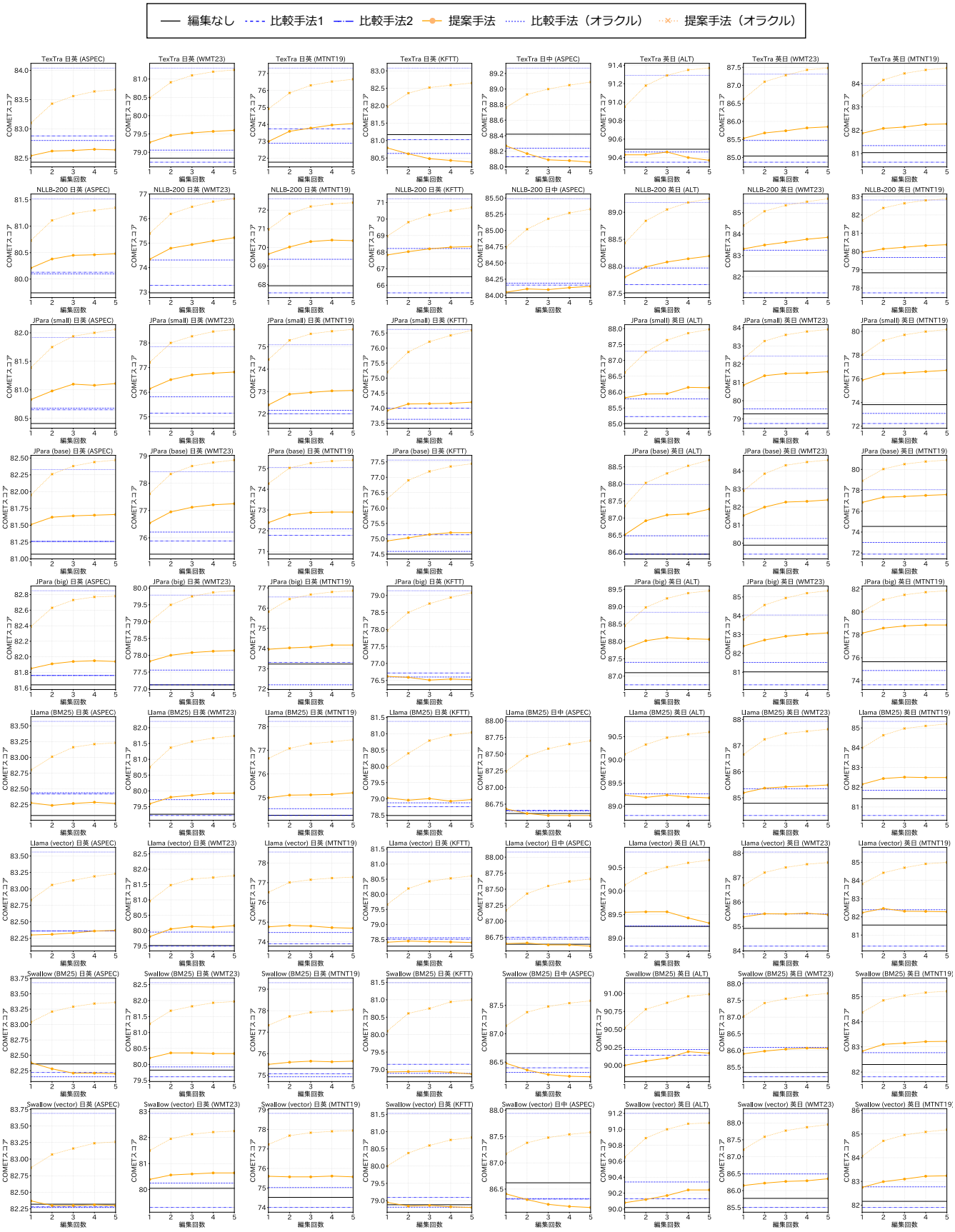

図 A·6 評価用データの誤り部分集合に対する COMET スコア: 編集回数ごとの変化

図 A·7 未編集の起点言語文に対する翻訳文と提案手法 ($k = 5$) によって得た翻訳文の文単位品質スコアの差分と COMET スコアの差分の対応