BART 文章校正モデルにおけるコピー機構の有用性の検証

北岡佑一 [1]　真嘉比愛 [1]
[1] ちゅらデータ株式会社
{y.kitaoka,a.makabi}@churadata.okinawa

概要

近年、大規模言語モデル（LLM）の発展により、高度な自然言語処理タスクの実現が可能となっている。しかし、文章校正タスクにおいて、LLM は入力文の意図を超えた過剰な書き換えを行う傾向があり、元の文意を保持できないという課題がある。

本研究では、BART モデルにコピー機構を導入することで、必要最小限の編集に限定した文章校正の実現を目指した。評価実験では、ERRANT による F0.5 スコアを用いて編集精度を測定し、従来の BART モデルと比べて精度が向上したのを確認した。また、LLM である Gemini とも比較を行い、本手法が有効であることを示した。

この成果は、我々の文章校正 AI サービス「ちゅらいと」の改善に向けた重要な知見となる。

1　はじめに

1.1　研究背景

文章校正タスクにおいて、LLM の活用が進んでいるものの、入力文に対して過剰な書き換えを行い、元の文意を大きく変更してしまう傾向がある。例えば、単純な誤字脱字の修正であっても、文全体を別の表現に書き換えてしまうケースが多く見られる。このような過剰な編集は、文章校正タスクにおいて望ましくない。

1.2　研究目的

本研究では、BART モデルにコピー機構を導入することで、必要最小限の編集に限定した文章校正の実現を目指す。これにより、入力文の意図を保持しながら、適切な誤り訂正を行うモデルの構築を目的とする。評価には ERRANT による F0.5 スコアを用い、編集の精度を定量的に測定する。

2　関連研究

Zhao ら (2019) [1] が提案したトランスフォーマーのコピー機構は、入力文から必要な情報を直接コピーすることで、出力の精度向上を図る手法である。基本的なアプローチとして、エンコーダ-デコーダの注意重みをコピー分布として使用し、生成確率を制御する方式が提案されている。文章校正タスクにおいては、コピー機構は入力文の大部分を保持しながら、必要な箇所のみを修正するために重要な役割を果たす (小川・山本 (2020) [2])。特に Transformer ベースのモデルにコピー機構を導入することで、過剰な書き換えを抑制し、最小限の編集に限定した校正が可能となる。また、要約タスクにおいても入力文の重要な単語を適切にコピーする手法が開発されている (長谷川ら (2020) [3]・Xu ら (2020) [4])。

3　モデルアーキテクチャ

3.1　ベースモデルアーキテクチャ

本研究では、事前学習済みの日本語 BART モデル（ku-nlp/bart-large-japanese）をベースモデルとして採用する。このモデルは、エンコーダ-デコーダ型の Transformer アーキテクチャを採用しており、日本語 Wikipedia の約 1,800 万文を用いて事前学習されている。

3.2　トランスフォーマーのコピー機構の実装

トランスフォーマーのコピー機構は、入力文から必要な情報を直接コピーすることで、過剰な書き換えを抑制する手法である。本研究では、デコーダの最終層のクロスアテンション重みをコピー分布として利用する。生成確率 P_{gen} とコピー確率 $(1 - P_{gen})$ の重み付け和により、最終的な出力確率を以下の式で計算する：

$$P(w) = (1 - P_{gen}) \cdot P_{copy}(w) + P_{gen} \cdot P_{vocab}(w)$$

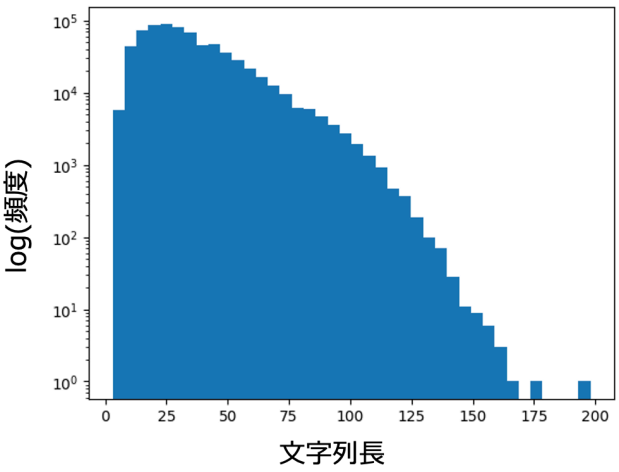

図 1　学習データの文字列長分布

ここで、$P_{vocab}(w)$ は語彙からの生成確率、$P_{copy}(w)$ はコピー確率を表す。コピー確率は以下の式で計算される：

$$P_{copy}(w) = \sum_{i:x_i=w} \alpha_{t,i}$$

$\alpha_{t,i}$ はデコーダの隠れ状態 s_t とエンコーダの隠れ状態 h_i の間のクロスアテンション重みを表している。この実装により、入力文の意図を保持しながら、必要最小限の編集に限定した文章校正が可能となる。今回このモデルを BART-copy と呼ぶことにする。

4　実験準備

4.1　学習データと前処理

今回の学習データとしては、田中ら [5] によって構築された「日本語 Wikipedia 入力誤りデータセット (v2)」(JWTD) を用いた。本研究では、編集前の文を入力文、 編集後の文をラベルとして使用した。データセットの統計情報は以下の通りである：

- 学習データ : 696190 件
- テストデータ : 5440 件
- ゴールドデータ : 1127 件

また学習データの文字列長の分布を図 [1] に示す。これにより、200 文字を超えるものはないとわかる。

以上のことより、JWTD データでは文章が短いので、一般的な評価として不適と考えた。よって、ゴールドデータを 1000 文字前後に拡張したデータを別途 1037 件用意した。こちらのデータがゴールドデータの 1127 件より減っているのは、1000 文字前後のデータを作成する際に、不自然な文章になったものを除いたためである。

前処理として、日本語 BART モデル（ku-nlp/bart-large-japanese）の要件に従い、Juman++による形態素解析を行った。

4.2　評価指標

本研究では、文章校正モデルの性能評価に ERRANT（ERRor ANnotation Toolkit）を採用する。ERRANT は、編集操作の正解と予測の間の一致度を測定するための評価指標である。ERRANT による評価では、以下の 3 つの指標を用いる：

- 適合率（Precision）：モデルが提案した編集のうち、正しい編集の割合
- 再現率（Recall）：正解の編集のうち、モデルが検出できた編集の割合
- F0.5 スコア：適合率と再現率の重み付き調和平均 (適合率優位)

F0.5 スコアは以下の式で計算される：

$$F_{0.5} = \frac{(1 + 0.5^2) \cdot (precision \cdot recall)}{0.5^2 \cdot precision + recall}$$

F0.5 スコアは適合率を再現率よりも重視する指標である。これは、文章校正タスクにおいて、誤った編集（偽陽性）を減らすことが、編集の見落とし（偽陰性）を減らすことよりも重要であるためである。つまり、不適切な編集によって文意が変わってしまうリスクを抑制することを重視している。よって今回は、F0.5 スコアを主要な評価指標として用いる。

5　実験結果

今回の実験では JWTD のデータを、学習データを学習に、テストデータを検証に、ゴールドデータをベンチマークとして用いた。また、Gemini 1.0 Pro、Gemini 1.5 Flash(それぞれ Few-shot) を比較対象として採用した。Few-shot では、以下のようなプロンプトを用いて文章校正タスクを実行した：

1. タスクの説明：
2. 例示（9 件）：
    - 入力文と校正後の文のペアを 9 つ提示
    - 各例は、JWTD データセットの拡張時に作成されたデータの一部
    - 校正の傾向が理解できるような典型的な例を使用

Gemini を採用した理由は、現在最も高性能な商用 LLM の一つであり、文章校正タスクにおける最新のベースラインとして適しているためである。特に

Gemini 1.5 Flash を選択したのは、高速な推論が可能で実用的なシステムの構築に向いているためである。

5.1 実験 1:JWTD の学習と推論結果

まず、学習した BART モデルと Gemini を比較した。実験結果を表 [1] に示す。

表 1 コピー有無でのゴールドデータの ERRANT 評価

	F0.5	Pre.	Rec.	TP	FP	FN
BART	0.6213	0.6270	0.5995	738	439	493
BART-copy	0.6435	0.6486	0.6239	**768**	416	**463**
BART-copy 改	**0.6479**	**0.6544**	**0.6231**	767	405	464
Gemini 1.0 Pro	0.5045	0.592	0.5145	621	616	586
Gemini 1.5 Flash	0.5106	0.3969	0.3969	479	**392**	728

BART-copy を導入することで、F0.5 スコアが 0.6213 から 0.6435 に向上した。特に、適合率 (Precision) が 0.6270 から 0.6486 に、再現率 (Recall) が 0.5995 から 0.6239 に改善している。これは、コピー機構により入力文から適切な単語を直接コピーすることで、過剰な書き換えが抑制されたためと考えられる。

アーキテクチャの初期値を調整 (BART-copy 改とする) をすることで、F0.5 スコアは 0.6479 まで向上し、誤った編集（FP）が 439 から 405 に減少した。一方で、正しい編集の検出と見落としについては、コピー機構導入時とほぼ同等の性能を維持している。

Gemini との比較では、Gemini 1.0 Pro と Flash の F0.5 スコアがそれぞれ 0.5045、0.5106 と、BART-copy を大きく下回る結果となった。

5.2 実験 2: 中文の推論実験

JWTD データの文字数分布を分析したところ、図 [1] より大半が 25 文字前後の文で構成されていることが判明した。そこで、本モデルの長文に対する性能を評価するため、100 文字以上の文 141 件を対象とした推論実験を実施した。実験結果を表 [2] に示す。

まず、実験 1 と比べて精度は大きく落ち込んでいる。さらに、Gemini と比べても再現率 (Recall) 以外は性能が低いことがわかる。しかし、コピー機構を導入したモデルでは F0.5 スコアが 0.3168 から 0.3585 へと向上し、再現率が 0.4267 から 0.4867 に改善した。アーキテクチャの改修により、F0.5 スコアは 0.3657 まで向上し、誤った編集（FP）が 151 から 139 に減少した。

表 2 100 文字以上のゴールドデータの ERRANT 評価

	F0.5	Pre.	Rec.	TP	FP	FN
BART	0.3168	0.2977	0.4267	64	151	86
BART-copy	0.3585	0.3364	**0.4867**	**73**	144	**77**
BART-copy 改	0.3657	0.3443	**0.4867**	**73**	139	**77**
Gemini 1.0 Pro	0.4533	0.4503	0.4658	68	83	78
Gemini 1.5 Flash	**0.4766**	**0.5044**	0.3904	57	**56**	89

5.3 実験 3: 長文の推論実験

表 3 長文に拡張したゴールドデータの ERRANT 評価

	F0.5	Pre.	Rec.	TP	FP	FN
BART-copy	0.0076	0.0062	0.0639	73	11642	1070
Gemini 1.0	**0.2524**	**0.2462**	**0.2939**	**179**	**548**	**430**
Gemini 1.5	0.2141	0.2086	0.2397	146	554	463

文章校正とは本来、長文内に間違いがあるかないかを検知するタスクであるので、ゴールドデータを 1000 文字前後に拡張したデータ (1037 件) でも推論の実験を行った。

表 [3] に実験結果を示す。BART-copy では、F0.5 スコアが 0.0076 と著しく低い性能となり、FP が 11,642 件と非常に多くなった。一方、Gemini は学習データの制約を受けないため、Gemini 1.0 Pro では F0.5 スコアが 0.2524 と比較的安定した性能を示した。

5.4 実験 4: コスト比較

生成速度の実験として、実験 1(1127 件) での BART-copy と Gemini の生成速度とコストを比較した。コストの比較においては、サーバーレス実行を想定した場合の実行時間あたりの処理コストと、API リクエストあたりの課金を別々に算出し、実際の運用シナリオに応じた比較を表 [4] で行った。単価の計算は GPU モデルの BART-copy(GPU) は AWS の EC2 インスタンスである g4dn.xlarge、BART-copy(CPU) は m5.xlarge を使用した際の料金を

表 4 処理時間とコストの比較

	処理時間 (分)	単価 ($/分)	コスト ($)
BART-copy(GPU)	**1.5**	0.00876	**0.013**
BART-copy(CPU)	8.3	**0.0032**	0.027
Gemini 1.0 Pro	26.2	0.65	17.1
Gemini 1.5 Flash	26.2	0.099	2.6

分単位で計算した。また、Gemini はそれぞれのプランにおける API リクエストあたりの課金を元に、処理時間を換算してコストを算出した。この結果から、ユースケースに応じて適切なモデルを選択する必要があることが分かる。ただし、Gemini はトークンベースの従量課金制を採用している点、BART モデルは EC2 インスタンスをホスティングする必要がある点で、単純なコスト比較は困難なことは注意が必要である。

6 考察

6.1 校正精度に関する考察

- 短文 (25 文字前後) : BART-copy 改が高い性能を示した。F0.5 スコアが 0.6213 から 0.6435 に向上し、特に誤った編集（FP）が 439 から 405 に減少したことは、入力文の意図を適切に保持できていると考えられる。
- 中文 (100 文字以上) : コピー機構の効果は維持されたが、性能は大きく低下した。F0.5 スコアは 0.3585 と、Gemini の 0.4766 を下回る結果で、特に FP の検出性能の低さが顕著。
- 長文 (1000 文字以上) : BART-copy の性能は著しく低下し、実用レベルでない。一方、Gemini は 0.2 程度の性能を維持できたが、これも実用的でない。

6.2 処理効率とコストに関する考察

モデルの特性により、以下のような使い分けが効果的であると考える。

1. BART-copy：
    - 短文処理に特化
    - 高速な処理（1127 件を 1.5 分で処理）
    - 経済的なコスト（CPU 環境で 0.0032$/分）
2. Gemini：
    - 長文でも安定した性能
    - トークンベースの従量課金
    - 処理速度は劣るものの、長文処理では信頼性が高い

これらの知見から、実用的な文章校正システムの実現には、文長に応じたモデルの使い分けと、長文処理に特化した学習データの整備が必要と考察する。

7 まとめ

7.1 研究成果

本研究では、文章校正タスクにおける BART-copy の有効性を検証した。まず BART-copy は、コピー機構を導入することで、入力文の意図を保持しながら過剰な書き換えを抑制し、必要最小限の編集を実現することに成功した。さらに、Gemini との比較実験を通じて、各モデルの特性が明らかになった。BART-copy は短文処理において優れた性能と経済性を示す一方、Gemini は処理速度では劣るものの、長文処理において安定した性能を発揮することが確認された。ただし、Gemini はトークンベースの従量課金制を採用しているため、単純なコスト比較は困難である。これらの知見から、実用的な文章校正システムを実現するためには、文長に応じて適切なモデルを使い分けること、および長文処理に特化した学習データの整備が必要であることが示唆された。

7.2 今後の課題

まず、長文処理の性能向上のため、文長分布を考慮したデータ収集と学習方法の改善が必要である。現在 1000 文字以上の長文データセットの構築を進めており、今後の実験での評価を行う予定である。

また、ルールベースの手法との組み合わせも検討すべきである。特に、固有名詞や専門用語などの処理において、ルールベースの確実性とニューラルネットワークの柔軟性を組み合わせることで、より堅牢な校正システムの構築が期待できる。現状の「ちゅらいと」では、ルールベースの処理を一部組み込んでおり、今後もルールベースの拡充を進める予定であるので、その成果を本研究に反映させることが可能である。

さらに、マルチタスク学習の導入も有効な改善策となりうる。文章校正タスクと関連する他のタスク（例：文法誤り検出）を同時に学習することで、モデルの汎化性能の向上が期待できる。

最後に、Gemini 2.0 や他の LLM との比較評価も必要である。特に、モデルサイズやコストパフォーマンスの観点から、実用的なシステムの構築に向けた最適なアプローチを見極める必要がある。これらの課題に取り組むことで、より実用的な文章校正システムの実現が可能となるだろう。

参考文献

[1] Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. **NAACL**, pp. 156- –165, 2019.

[2] 小川耀一朗, 山本和英. 日本語文法誤り訂正における誤り傾向を考慮した擬似誤り生成. 言語処理学会第26回年次大会発表論文集, pp. 505–508, 2020.

[3] 上垣外 英剛長谷川 駿, 奥村学. 教師有りコピー機構を用いた要約文生成. 人工知能学会全国大会 (第 34 回) 人工知能学会全国大会論文集, 2020.

[4] Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. Self-attention guided copy mechanism for abstractive summarization. **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 1355–1362, 2020.

[5] 田中佑, 村脇有吾, 河原大輔, 黒橋禎夫. 日本語wikipediaの編集履歴に基づく入力誤りデータセットと訂正システムの構築. 自然言語処理, Vol. 28, No. 4, pp. 995–1033, 2021.