言語モデルを用いた漢文の返り点付与と書き下し文生成

王昊　清水博文　河原大輔
早稲田大学理工学術院
{conan1024hao@akane.,bowen1205@toki.,dkw@}waseda.jp

概要

漢文は約 2000 年前の弥生時代に中国から日本に伝えられ，それ以降日本文学に多大な影響を与えた．今でも大学入学共通テストの国語において漢文は 200 点の内 50 点を占めている．そんな中，中国にある豊富な言語資源に比べ，日本にある漢文の書き下し文資源が非常に少ない．この問題を解決するために，本研究は世界初の漢文訓読データセットを構築する．そして，漢文理解において重要視される返り点付与，書き下し文生成の二つのタスクに対し，言語モデルを用いて精度向上を試みる．また，人間の評価結果と比較することで，最適な自動評価指標について議論する．

1　はじめに

漢文は弥生時代 [1] に中国から日本に伝えられた．そして奈良時代 [2] から日本語の文章として読めるよう，漢文の語順構成を維持しながら訓点を付ける漢文訓読と，日本語の文体として書き直した漢文訓読文 (または書き下し文) が発明された．漢文は『万葉集』[3] や『源氏物語』[4, 5] など，多くの日本文学作品に影響を与えた．今でも漢文は大学入学共通テストの国語において 200 点の内 50 点を占めており，漢文が日本文化に及ぼしている影響の大きさを示している．

中国語と日本語は共通の漢字を多く持つが，日本人にとって漢文を読むことは容易ではない．中国語，そして漢文の語順は，英語と同じ SVO (主語-動詞-目的語) である一方，日本語の語順は SOV (主語-目的語-動詞) である．そして，中国語は孤立語であり，時制や格などによって語の形は変化しない．一方，日本語は膠着語であり，接頭辞や接尾辞のような形態素がその語の文法関係を示している．漢文を SVO から SOV に，孤立語から膠着語に変換するために，日本人は句読点，返り点，送り仮名などからなる漢文訓読システム [6] を開発した．

表 1　漢文訓読データセットの例．データは白文，日本語読み順 (数字は白文中の順位を表している)，書き下し文の 3 つ組から構成される．

白文	日本語読み順	書き下し文
春眠不覚暁	12543	春眠暁を覚えず
処処聞啼鳥	12453	処処啼鳥を聞く
夜来風雨声	12345	夜来風雨の声
花落知多少	12345	花落つること知んぬ多少ぞ

中国には豊富な漢文の言語資源があるが，日本におけるそれらの書き下し文データは極めて少ない．例えば，『全唐詩』には 48,900 首以上の唐詩が収録されており，その全てにオンラインでアクセス可能である．しかし，我々の知る限りでは，日本では約 500 首の唐詩の書き下し文データしかインターネット上に存在しない．この大きなギャップは，日本における漢文研究及び漢文教育の阻害要因となっている．『漢文大系』などの書籍の中に書き下し文のデータは多く存在するが，それらに OCR を適用し，データを整形するには膨大なコストがかかる．そのため，高性能な書き下し文生成器を構築することが書き下し文資源の不足を解消する最も効率的な方法と考えられる．

従来の研究 [7, 8, 9, 10, 11] では，返り点付与と書き下し文生成を含む一連の漢文に関する言語処理の手法が提案された．しかし，ルールベースであり性能が不十分である上，これらの研究ではデータセットを作っておらず，定量的評価を行っていない．本研究では，世界初の白文[1)]-書き下し文ペアからなる漢文訓読データセットを構築する (表 1)．これを基に，言語モデルを用い，返り点付与器と書き下し文生成器を構築し，両タスクにおいて定量評価を行う．そして，書き下し文の生成結果に対し，人手評価の結果と比較することで，最も適切な自動評価指標について議論する．さらに，パイプラインを構築し，白文の事前ソート (返り点付与) が書き下し文生成に貢献するかどうかを検証する．

1)　句読点，返り点，送り仮名が付いていない漢文のこと．

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

表 2　データセットの統計量情報 (白文パート).

Split	詩の数	文の数	文字数
Train	372	2,731	16,411
Validation	46	320	2,038
Test	47	370	2,254

2　関連研究

UD-Kundoku [9, 10] は Universal Dependencies [12] に基づく漢文を書き下し文に変換する encode-reorder-decode モデルである．まず encode では白文に対し形態素解析を行い，依存構造解析を行う [7]．reorder ではルールベースによる返り点付与を行い，白文を日本語語順にソートする．最後の decode ではルールベースによる送り仮名の付与を行う．これらの研究では書き下し文の生成結果に対して BLEU [13] と RIBES [14] を用いて評価を行っているが，数サンプルの評価に留まっており，定量的評価を行っていない．

3　データセット構築とタスク設計

本研究は白文，日本語読み順，書き下し文の 3 つ組から構成される漢文訓読データセットを構築する．漢文で使われる語彙や文法は時代によって変化するため，多くの時代をカバーするテキストを収集することが望ましいが，現時点ではそのような包括的なデータセットを作ることは困難である．本研究では，書き下し文が付与されているデータソースとして最も大きいと考えられる『唐詩選』[2] を基にデータセットを作成する．前処理として，ルールベースで書き下し文から白文の日本語語順を抽出し，置き字や再読文字など特殊な文字に対し人手でアノテーションを行う．また，辞書を用いて旧字体を可能な限り新字体に変換し，言語モデルの OOV 問題を軽減する．一つの詩の中の文が分割されないように，本研究では GroupShuffleSplit を使ってデータセットを Train，Validaiton，Test に分割した．表 1 にデータの例，表 2 にデータセットの統計量情報を示す．

構築したデータセットに基づき，返り点付与と書き下し文生成の二つのタスクを設定する．返り点付与 [3] では，白文を日本語の読み順 (SVO から SOV) に変換する．書き下し文生成は，白文を書き下し文に変換する seq2seq タスクである．

2)　https://kanbun.info/syubu/toushisen000.html
3)　実質上は文字の並べ替えとなっている．

4　実験設定

本研究は事前学習モデルをファインチューニングすることで，返り点付与と書き下し文の二つのタスクを解く．実験で使用する事前学習モデル及びハイパーパラメータを付録 A と B に示す．

4.1　タスクの実装

本節では，返り点付与と書き下し文のそれぞれの実装を説明する．本研究ではパイプラインも構築し，白文の事前ソート (返り点付与) が書き下し文生成に貢献するかどうかについて実験を行う．図 1 にパイプラインの概要を示す．

返り点付与に対して，本研究では BERT-like モデル [15, 16, 17, 18] を用いた rank-based のソート手法を採用する．入力としては，白文を文字単位に分割し，それぞれの文字を { 文字 }{ 白文中の文字の順位 }[SEP]{ 白文 } の形に変換する．白文中の文字の順位は同じ文字が二つ以上出現する場合に対応するために追加している．それぞれの文字の日本語語順中の順位を白文の長さで正規化し (長さ 5 の文に対し，順位は 1, 2, ..., 5 から 0.2, 0.4, ..., 1 に正規化される)，学習における正解の値として設定する．モデルが出力した順位を昇順にソートし，元の文字に戻せば，日本語語順の白文を得ることができる．図 1 の (A) に我々のソート手法の様子を示している．

書き下し文生成に対しては，T5 [19] と GPT [20] を用いて，白文から書き下し文を生成する．各モデルの真の性能を見るため，生成結果に対しては一切フィルタリングを行わない．

パイプラインについては，返り点付与モデルによって日本語語順にソートされた白文を T5 または GPT に入力し，事前の返り点付与が書き下し文生成に貢献するかどうかを検証する．

4.2　自動評価指標

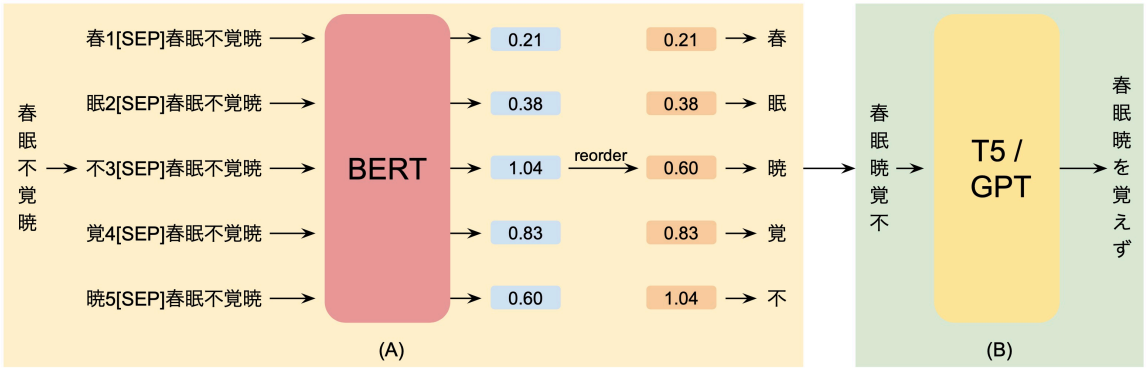
語句の並べ替えの研究 [21, 22, 23] に従い，本研究では以下の指標を用いて自動評価を行う．

Kendall's Tau (τ)　この指標は，二つの文の順位相関を測るものである．予測された文字順を真の文字順に並べ替えるために必要な交換回数が少ないほど，相関が強く，性能が良いことを意味する．

$$\tau = 1 - \frac{4(\#inversions)}{\#char(\#char - 1)}$$

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

図 1　パイプラインの概要図. (A) は返り点付与器，(B) は書き下し文生成器である.



Perfect Match Ratio (PMR)　予測された文字順と真の文字順が完全に一致する割合を示す指標.

書き下し文の評価について系統的な研究がないため，先行研究で用いられた BLEU [13] と RIBES [14] に加え，ROUGE-L [24] と BERTScore [25] を用いて自動評価を行う．単語ベースの評価は形態素解析への依存度が高く，また，漢文の関連パッケージが成熟していないことから，本研究では文字ベースで上記の指標を計算する．BERTScore については，文字ベースの東北大 BERT[4] を用いて計算する.

4.3　人手によるアノテーション

本研究は中国語と日本語のバイリンガル三名にアノテーションを依頼する．アノテータの選考基準としては以下のものを設ける: (1) 中国語の白文を読めること．(2) 大学入学共通テストの漢文パートで満点が取れること.

返り点付与に対しては，返り点付与器と人間の性能を比較するために，同じソート作業を参考文献やインターネットアクセスなしで行ってもらう．結果を集計し，Kendall's Tau と PMR スコアを計算する.

書き下し文生成に対しては，以下の三つの指標で 1 から 5 の 5 段階で評価してもらう．書き下し文の正しさを評価したいため，この評価では参考文献の参照を可能としている．また，データセットの品質を測るために，正解の書き下し文も人手評価する.

Relevance　白文を不足なく，乖離なく書き下しているかどうか.

Accuracy　日本語としての語彙的，文法的正しさ．読み順の正しさ.

Fluency　文の流暢さ，自然さ．漢詩のリズム感が残っているかどうか.

4)　https://huggingface.co/cl-tohoku/bert-base-japanese-char-v2

表 3　返り点付与における Kendall's Tau (τ) と PMR の評価結果. UD-Kundoku はベースラインで, Human スコアは三人のアノテータの結果の平均である.

Model Setup	τ	PMR
UD-Kundoku	0.770	0.402
Human	0.844	0.606
BERT-japanese-char	0.898	0.637
RoBERTa-japanese-char-wwm	0.894	0.600
BERT-chinese	0.917	0.689
RoBERTa-chinese-wwm-ext	0.920	0.718
RoBERTa-classical-chinese-char	**0.944**	**0.783**

5　結果と考察

5.1　返り点付与

返り点付与の評価結果を表 3 に示す．全ての BERT-like モデルはベースラインより高い精度を達成した．中国語のモデルは，日本語のモデルよりも若干良い精度を示し，漢文コーパスで事前学習した RoBERTa-classical-chinese-char は最も良い精度を示した．これは，事前学習で使われているコーパスにおける漢文の割合によるものだと考えられる.

人間と RoBERTa-japanese-char-wwm の PMR スコアはほぼ同じであるが，Kendall's Tau はモデルの方が 5.9%高い．これは，細かい部分の予測では人間よりも BERT の方が精度が高いことを示している.

5.2　書き下し文生成

書き下し文生成に対する評価結果を表 4 に示す.

自動評価では，全モデルが全ての評価指標でベースラインを上回った．mT5 の性能はモデルサイズの増大につれて上昇し，mT5-large が最も良い性能を示した．mGPT と mT5-small の性能はお互いに近い.

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

表 4 書き下し文生成の評価結果. UD-Kundoku はベースラインである.

Model Setup	BLEU	RIBES	ROUGE-L	BERTScore	Relevance	Accuracy	Fluency
UD-Kundoku	0.097	0.309	0.546	0.884	-	-	-
正解書き下し文	-	-	-	-	4.958	4.951	4.949
mT5-small	0.317	0.428	0.659	0.914	3.219	3.002	3.153
mT5-base	0.462	0.520	0.735	0.930	-	-	-
mT5-large	**0.514**	**0.583**	**0.747**	**0.934**	**3.948**	**3.884**	**3.904**
mGPT	0.303	0.476	0.606	0.898	2.548	2.270	2.236

表 5 自動・人手評価指標の Pearson (r), Spearman (ρ) 相関係数. 人手評価指標間の相関も示している.

Metric	Relevance		Accuracy		Fluency	
	r	ρ	r	ρ	r	ρ
BLEU	0.667	0.650	0.637	0.605	0.594	0.576
RIBES	0.480	0.497	0.453	0.449	0.389	0.417
ROUGE-L	0.688	0.677	0.631	0.610	0.599	0.584
BERTScore	**0.707**	**0.691**	**0.671**	**0.642**	**0.644**	**0.625**
Relevance	-	-	0.862	0.849	0.835	0.829
Accuracy	0.862	0.849	-	-	**0.946**	**0.947**
Fluency	0.835	0.829	**0.946**	**0.947**	-	-

人手評価では，mT5-small，mT5-large，mGPT の生成結果のみをアノテータに評価させた．自動評価と同様，mT5-large が最も良い性能を示した．一方，mGPT は mT5-small の 8 倍以上のパラメータ数を持つが (詳細なモデルサイズを付録 A に示す), mT5-small のスコアは大きく mGPT を上回った．mT5，mGPT は共に主に mC4 [19] で事前学習しているため，コーパスの影響はほぼ排除できる．推測の一つとしては，mT5 のエンコーダが漢文の理解に大きな役割を果たしていると考えられる．しかし，これは仮説であり，追加の実験が必要である．正解書き下し文は非常に高いスコアを獲得し，本データセットの書き下し文データが高品質であることが証明された．

表 5 に自動評価指標と人手評価指標間の相関係数を示す．BERTScore は三つの人手評価指標全てと最も高い相関を示した．ランクに基づく RIBES は最も低い相関を示した．BLEU と ROUGE-L に比べ，BERTScore は Relevance でわずかにリードしているが，Accuracy と Fluence では大幅に優れていた．これは，BERTScore が潜在的な文単位の意味を捉えることができる [25] ため，文の正しさと流暢さを判断することができたと推測する．

人手評価指標間の相関も表 5 に示している．Accuracy と Fluency の相関が最も大きく，語彙的，文法的に正しい日本語は流暢であることがわかる．

表 6 パイプラインの評価結果. 一行目は素の生成結果，二行目は RoBERTa で事前ソートしてから生成させたもの，三行目は正解でソートしてから生成させたもの.

Model Setup	BLEU	RIBES	ROUGE-L	BERTScore
mT5-small	0.317	0.428	0.659	0.914
+ reorder	0.328	0.420	0.701	0.916
+ reorder (gold)	**0.359**	**0.451**	**0.727**	**0.919**
mT5-base	**0.462**	0.520	0.735	0.930
+ reorder	0.413	0.486	0.735	0.926
+ reorder (gold)	0.461	**0.529**	**0.770**	**0.932**
mT5-large	**0.514**	**0.583**	0.747	0.934
+ reorder	0.479	0.551	0.748	0.931
+ reorder (gold)	0.502	0.573	**0.774**	**0.935**
mGPT	0.303	0.476	0.606	0.898
+ reorder	0.303	0.467	0.612	0.894
+ reorder (gold)	**0.340**	**0.508**	**0.642**	**0.900**

5.3 パイプライン

パイプラインの評価結果を表 6 に示す.

返り点付与モデル (RoBERTa) で事前ソートした結果，mT5-small のスコアが改善されたが，mT5-base と mT5-large はほとんどの評価指標で性能の低下を示した．生成モデルの性能が上がれば，徐々に語順を考慮しながら書き下し文を生成できるようになると推測する．返り点付与モデルの誤った予測により生成モデルが混乱し，正しい語順を予測できなくなったと考えられる．

一方，正解を用いた事前ソートは，ほぼ全てのモデルにおいて一定の性能向上を示した．これは，正しい返り点付与は書き下し文の生成に貢献することを示している．

6 まとめと今後の展望

本研究は，書き下し文資源の不足を解消するために，漢文訓読データセットを構築し，言語モデルを用いた返り点付与と書き下し文生成を試みた．今後は，データセットを継続的に更新し，より包括的な漢文テキストを含むよう改良したいと考える．また，漢文の専門家と協力し，書き下し文生成結果の評価についてより適切な指標を探求していきたい．

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

謝辞

本研究は JSPS 科研費 JP21H04901 の助成を受けて実施した.

参考文献

[1] 沖森卓也. 日本語全史. 筑摩書房, 2017.

[2] 金文京. 漢文と東アジア－訓読の文化圏. 岩波書店, 2010.

[3] 小林芳規. 万葉集における漢文訓読語の影響. 国語学, No. 58, pp. 23–47, 1964.

[4] 段笑曄. 『源氏物語』における『白氏文集』引用の特色–登場人物の口ずさんだ詩句をめぐって. 北陸大学紀要 = Bulletin of Hokuriku University, No. 32, pp. 181–192, 2008.

[5] 長瀬由美. 源氏物語と平安朝漢文学. 勉誠出版, 2019.

[6] Sydney Crawcour. **An introduction to Kambun**. University of Michigan, 1965.

[7] 安岡孝一. 漢文の依存文法解析と返り点の関係について. 日本漢字学会第 1 回研究大会予稿集, pp. 33–48, 2018.

[8] 安岡孝一. Universal dependencies treebank of the four books in classical chinese. **DADH2019: 10th International Conference of Digital Archives and Digital Humanities**, pp. 20–28, 2019.

[9] 安岡孝一. 漢文の依存文法解析にもとづく自動訓読システム. 日本漢字学会第 3 回研究大会予稿集, pp. 60–73, 2020.

[10] 安岡孝一. 漢文自動訓読ツール ud-kundoku の開発. 東洋学へのコンピュータ利用 第 32 回研究セミナー, pp. 3–25, 03 2020.

[11] 安岡孝一, ウィッテルンクリスティアン, 守岡知彦, 池田巧, 山崎直樹, 二階堂善弘, 鈴木慎吾, 師茂樹, 藤田一乗. 古典中国語（漢文）universal dependencies とその応用. 情報処理学会論文誌, Vol. 63, No. 2, pp. 355–363, 2022.

[12] Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. Universal Dependencies. **Computational Linguistics**, Vol. 47, No. 2, pp. 255–308, June 2021.

[13] 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**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[14] 平尾努, 磯崎秀樹, Kevin Duh, 須藤克仁, 塚田元, 永田昌明. Ribes: 順位相関に基づく翻訳の自動評価法て. 言語処理学会年次大会発表論文集, Vol. 17, pp. D5–2, 2011.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[16] 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. arXiv, 2019. abs/1907.11692.

[17] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. arXiv, 2019. abs/1909.11942.

[18] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. arXiv, 2020. abs/2006.03654.

[19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, Vol. 21, No. 140, pp. 1–67, 2020.

[20] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[21] Baiyun Cui, Yingming Li, and Zhongfei Zhang. BERT-enhanced relational sentence ordering network. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6310–6320, Online, November 2020. Association for Computational Linguistics.

[22] Pawan Kumar, Dhanajit Brahma, Harish Karnick, and Piyush Rai. Deep attentive ranking networks for learning to order sentences. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, No. 05, pp. 8115–8122, Apr. 2020.

[23] Yutao Zhu, Jian-Yun Nie, Kun Zhou, Shengchao Liu, Yabo Ling, and Pan Du. Bert4so: Neural sentence ordering by fine-tuning bert. arXiv, 2021. abs/2103.13584.

[24] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In **Text Summarization Branches Out**, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[25] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In **International Conference on Learning Representations**, 2020.

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

A　事前学習モデル

以下に実験に用いた事前学習モデルの詳細を示す．BERT-like モデルは表 7，生成モデルは表 8 に示す．

表 7　事前学習モデルの詳細 (返り点付与).

model	corpus	#dimension	#layers	#heads	vocabulary size
BERT-japanese-char (cl-tohoku/bert-base-japanese-char-v2)	Wikipedia(ja)	768	12	12	6,144
RoBERTa-japanese-char-wwm (ku-nlp/roberta-base-japanese-char-wwm)	Wikipedia(ja) + CC-100(ja)	768	12	12	18,377
BERT-chinese (bert-base-chinese)	Wikipedia(zh)	768	12	12	21,128
RoBERTa-chinese-wwm-ext (hfl/chinese-roberta-wwm-ext)	Wikipedia(zh) + ext	768	12	12	21,128
RoBERTa-classical-chinese-char (KoichiYasuoka/roberta-classical-chinese-base-char)	Wikipedia(zh) + *Daizhige* + ext	768	12	12	26,318

表 8　事前学習モデルの詳細 (書き下し文生成).

model	corpus	#params	#dimension	#layers	#heads	vocabulary size
mT5-small (google/mt5-small)	mC4 (101 languages)	172M	512	8	6	250,112
mT5-base (google/mt5-base)	mC4 (101 languages)	390M	768	12	12	250,112
mT5-large (google/mt5-large)	mC4 (101 languages)	973M	1024	24	16	250,112
mGPT (sberbank-ai/mGPT)	Wikipedia + mC4 (both 60 languages)	1,417M	2048	24	16	100,000

B　ハイパーパラメータ

表 9 に実験に用いたハイパーパラメータを示す．中括弧内の数字でグリッドサーチをして最適なものを選んでいる．

表 9　実験に用いたハイパーパラメータ.

Hyper-parameter	Value
learning rate	{1e-5, 2e-5, 5e-5}
batch size	{8, 16, 32}
epoch	{1-20}(BERT), {10, 20, 30}(T5), {1, 2, 3}(GPT)

C　書き下し文の生成例

表 10 に書き下し文の生成例を示す．mT5-large は最も性能がよく，正解と同じ生成結果を得ている．mT5-base と mT5-small も正解に近い結果を生成しているが，若干の誤りがある．mGPT は白文にある文字を繰り返すことがあり，人手評価のスコアが低い原因となっている．三列目の “未” は再読文字であり，“未だ... ず” と二回読む必要がある．これに対し，mT5-base と mT5-large は正しく書き下し文を生成しているが，mT5-small と mGPT は失敗している．

表 10　書き下し文の生成例.

Model Setup			
白文	投筆事戎軒	駆馬出関門	鳳林戈未息
正解書き下し文	筆を投じて戎軒を事とす	馬を駆って関門を出づ	鳳林戈未だ息まず
mT5-small	筆を投じて戎軒を事す	馬を駆って関門に出づ	鳳林戈未だ息し
mT5-base	筆を投じて戎軒に事す	馬を駆って関門に出で	鳳林戈未だ息まず
mT5-large	筆を投じて戎軒を事とす	馬を駆って関門を出づ	鳳林戈未だ息まず
mGPT	筆を投じて戎軒に事とすを事	馬を駆って関門を出でんとすも出で	鳳林戈未だ息まずかとすかとす鳳

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