大規模言語モデルを用いたシフト還元型句構造解析

中根 稜介 [1]　前川 在 [1]　上垣外 英剛 [2]　平尾 努 [3]　奥村 学 [1]
[1] 東京科学大学　[2] 奈良先端科学技術大学院大学　[3] 金沢大学
{nakane,maekawa,oku}@lr.pi.titech.ac.jp　kamigaito.h@is.naist.jp
hirao@se.kanazawa-u.ac.jp

概要

大規模言語モデル (LLM) は言語生成のみならず，さまざまな NLP タスクにおいて良好な結果を残している．本稿では，談話構造解析の一つである修辞構造解析において顕著な結果を残した，LLM によるシフト還元型解析法を拡張した句構造解析法を提案する．提案法は，シフト還元型解析でありながら，明示的にスタックとキューを持たず，解析位置をあらわすタグとその左右のテキストでそれを代替する．提案手法を，Penn Treebank (PTB) を用いて訓練し，PTB，Multi-domain Constituent Treebank (MCTB) を用いて評価した結果，従来の LLM を用いた seq2seq モデルによる解析法よりもドメインの違いに頑健であり，どのような文長に対しても安定して高い性能であることを確認した．

1　はじめに

句構造解析は自然言語処理の基盤タスクの一つであり，古くから研究が続けられている [1, 2]．句構造木では終端記号は単語，非終端記号はそれが支配する構成素 (句) の役割をあらわす．図 1 に例を示す．Wall Street Journal に句構造のアノテーションを与えた Penn Treebank (PTB) [3] をベンチマークデータセットとして，これまでに様々な解析手法が提案されてきた．初期の研究では，人間が考案した特徴を用い，CKY 法 [4]，シフト還元法 [5] などといった句構造木を導出するためのアルゴリズムが利用されていた．近年では，特徴抽出に単語埋め込みベクトルに基づくエンコーダを利用することで，これらの手法の性能は大きく改善された [6, 7]．

一方，LSTM や Transformer などのエンコーダデコーダモデルを用いて入力系列を別の系列へと変換する sequence-to-sequence (seq2seq) モデルは，いわゆる機械翻訳や自動要約などの言語生成タスクで顕著な結果を残した [8, 9]．seq2seq モデルはこうした言語生成タスクだけでなく，構文解析，特に句構造解析でも顕著な結果を残している [10, 11]．句構造木は線形化することで S 式として表現できることから，入力文，つまり単語列を S 式へ変換する問題として捉えることができるからである．こうした背景のもと，デコーダを大量のテキストデータで事前学習した大規模言語モデル (LLM) を用いて句構造木の S 式を予測する手法も提案されている [12]．

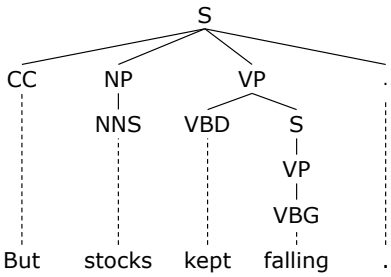

図 1　"But stocks kept falling." に対する句構造木 (Penn Treebank WSJ_2300 より)．なお，解析器の学習には等価な 2 分木を用いており，品詞タグは利用していない．

現在，ニュース記事のドメインである PTB での最高解析性能は F 値 96 を超えており非常に高い．一方，PTB で学習した最先端の解析器を最近公開された，対話，オンラインフォーラム，文学作品など様々なドメインで構成されるベンチマークデータセット (Multi-domain Constituent Treebank: MCTB) [13] で評価すると，F 値は 80-90 程度にとどまる [14, 15]．つまり，解析器にはまだまだ改善の余地があることが示唆されており，句構造解析という研究課題に取り組む価値は依然十分にある．

本稿では，文献 [12] 同様，LLM を用いるものの，線形化した構文木を予測するのではなく，シフト還元型解析法におけるシフト還元動作を予測する手法を提案する．動作予測という点では文献 [16] と同様であるが，明示的にスタック，キューを与えることなく，タグを用いて解析位置をあらわし，解析済みの単語列は S 式としてあらわす点が大きく異なる．

提案法を seq2seq モデルによる解析器と比較したところ，PTB，MCTB の双方において提案法が優れていることを確認した．提案法はドメインの違いに頑健であり，入力文の長さに関わらず高い解析性能を保持するという利点が明らかとなった．

2 関連研究

Bai ら [12] は，LLM により，線形化した構文木を予測する句構造解析法を提案した．この手法は，LSTM や Transformer による seq2seq モデルを LLM に置き換えたものであり，LLM の持つ言語生成能力を活かした手法である．線形化法として，いわゆる単純な S 式，シフト還元型解析における履歴，部分木 (スパン) の役割のアノテーションを比較したところ，単純な S 式を用いた場合の性能が最も良かったことを報告している．LLM として LLaMA-65B を採用した場合，PTB における F 値は 95.9 であり，最高性能の解析器とほぼ同等の性能を達成している．

我々の知る限り LLM を用いた句構造解析の研究は Bai らの手法 [12] のみであるが，構文解析と似たタスクである談話構造解析においてはいくつか LLM を用いた解析法が提案されている．Shen ら [17] は，GoLLIE [18] に基づき係り受けのアノテーションガイドラインを定義し，それを LLM に与えることで談話依存構造解析における文の係り元，係り先，それらの間の関係の同定に利用した．Thompson ら [19] は，SDRT [20] の解析において係り受け関係にある EDU (節に相当する談話構造の基本単位) の組とその間の関係を，LLM を用いて逐次的に決定する手法を提案した．これら 2 つの研究は係り受け関係にある 2 つの単位の同定とその間の談話関係の予測を，LLM を用いて行うものであり，LLM は生成というよりむしろ分類問題を解いている点でユニークである．Maekawa ら [16] は，句構造解析の談話構造版ともいえる修辞構造解析 [21] において，シフト還元法による解析を，LLM を用いて行う手法を提案した．プロンプトとしてスタックの上 2 つの部分木に相当するテキストスパン，キューの先頭に相当する EDU を与え，シフトか還元かを LLM に予測させて木を構築する．この手法も，LLM は生成というよりむしろ分類問題を解いており，古くからある構文解析法を LLM で模倣していると捉えられる．これらの手法は，談話構造解析を対象としているため，入力が文書となり非常に長い．よって，LLM に与える文脈は局所的にならざるを得ない．たとえば，Thompson らは 15 発話分の文脈しかみていない．Maekawa らはスタックの上 2 つのテキストスパンとキューの先頭の EDU しかみていないため，解析の初期段階では局所的な文脈しか考慮できない．

3 提案手法

本研究では，修辞構造解析のために提案された Maekawa らの LLM を用いたシフト還元型解析法 [16] を句構造解析のために拡張する．主な相違点は，スタック，キューを明示的に与えず，文全体と<head>というタグを用いてシフト還元動作を推定する点，還元動作を経て得られた部分木は S 式として表現する点である．これらにより，文全体を文脈として考慮できるうえ，すでに構築した部分木もシフト還元動作の推定に活用できる．

3.1 シフト還元型解析

シフト還元型解析は古くから知られる構文解析手法である [5]．解析済みの部分木を格納するスタック，これから解析対象とする単語を格納するキューを用いて，以下のシフト，還元動作を繰り返し適用して木を構築する．

シフト　キューの先頭の単語を取り出し，スタックに積む，

還元 (binary)　スタックの上 2 つの部分木を取り出し，それらを親となる非終端記号でマージして 1 つの部分木にし，スタックに積む，

還元 (unary)　スタックの一番上の部分木を取り出し，その親となる非終端記号を割り当てる．

動作として還元が選択された場合，親となる非終端記号を決定する分類問題を引き続き解くこととなる．すなわち，非終端記号数は 28 であるため，LLM は最初に 3 値分類問題を解き，選ばれた動作が還元の場合には 28 値分類問題をつづけて解く．なお，この操作を経て構築される句構造木は 2 分木となる．[1)]

3.2 プロンプト

LLM の入力には解析対象となる文を以下のプロンプトとして与える．

「スタック」 <head> 「キュー」

ここで，スタックはシフト後の単語あるいは還元動作後の部分木を格納する．部分木は S 式として格

1) 後述の評価の際には，2 分木を，ルールを用いて等価な多分木へと変換する．

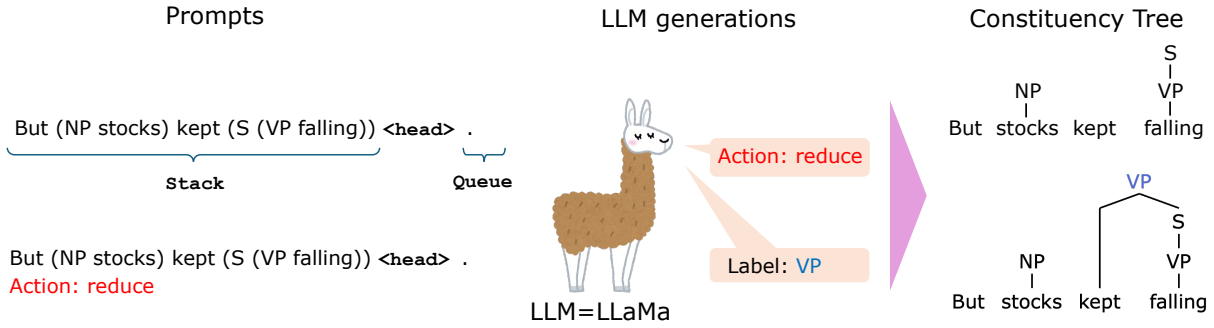

図 2 提案法のプロンプトと木の構築過程．スタックには下から順に"But"，部分木:"NP-stocks"，"kept"，部分木:"S-VP-falling", キューには"."が格納されている状態で LLM が reduce を選択した場合，"kept" と 部分木:"S-VP-falling" がマージされ VP 句 (部分木) が形成される．

納する．キューは未解析の単語列を格納する．ただし，LLM にはこれらを明示的にスタック，キューとして扱うように与えているわけではないことに注意して欲しい．<head>が解析位置をあらわすタグであることから，その左側がスタック，右側がキューとなることは自明である．[2] ただし，図 2 に示すように，スタック，キューとも単なる単語と部分的な S 式の系列として表現されているだけで，何番目の要素であるかという情報もない．この点，Maekawa らの手法 [16] とは異なることに注意されたい．また，解析中のすべての状態において文全体がスタックあるいはキューに格納される形で参照でき，すでに解析して構築された部分木も参照できる点でも異なる．

入力されたプロンプトに対し，LLM は以下の解析動作のいずれかを出力する．

```
Action: shift or reduce or unary
```

ここで reduce は還元 (binary)，unrary は還元 (unary) をあらわす．reduce あるいは unary が選ばれた場合には，入力プロンプトの 2 行目に，選択した動作を追加して非終端記号を予測する (図 2 参照)．

4 実験設定

4.1 データセット

解析器の訓練には PTB，評価には PTB に加え MCTB も用いた．PTB の訓練/開発/テストデータは，公式の分割設定に従い，それぞれ 39832 文，1700 文，2416 文である．MCTB は対話，オンラインフォーラム，法律文書，文学作品，レビューの 5 つのドメインの文に句構造のアノテーションを与えたデータセットであり，各ドメイン 1000 文である．

4.2 比較した解析手法

提案法の有効性を確認するため，Bai ら [12] による LLM を用いた seq2seq モデルの解析器，PTB における現在の世界最高性能を達成した Tian らの手法 [14]，その元となった Kitaev らの手法 [15] との比較評価を行った．Tian ら，Kitaev らの手法は，Transformer エンコーダ (前者が BERT [22]，後者が XLNet [23]) を特徴抽出に使ったチャート法を元にした解析法である．Bai らの手法に関しては，論文に掲載されたスコアと，我々と同等の設定で実験した結果の双方を比較した．

4.3 学習・評価の設定

提案法と seq2seq モデルの LLM には，unsloth の 4bit 量子化がなされた LLaMa3-8B [24][3] を用い，QLoRA [25] により追加学習を行った．訓練には PTB の訓練データを利用し，モデル選択，ハイパーパラメタ選択には，PTB の開発データを用いた．モデル選択の基準は開発データにおける損失である．学習の詳細については付録 A を参照されたい．

句構造木の評価には，予測された木と正解の木の間で一致するラベル付きスパンの割合を計算する PARSEVAL [26] 指標を用いる．なお，この計算にはデファクトスタンダードなツールである Evalb[4] を用いた．

5 結果と考察

Evalb を用いた評価結果 (F 値) を表 1 に示す．表中，Ours と S2S (ours) が我々の手元での実験結果であり，それ以外はそれぞれの論文に掲載されたスコアである．S2S (ours) と S2S-7B (Bai) は，我々が

2) 初期状態は"<head>文=単語列"であり，最終状態は"句構造木=S 式<head>"となる．

3) https://huggingface.co/unsloth/llama-3-8b-bnb-4bit
4) https://nlp.cs.nyu.edu/evalb/

表 1　PTB，MCTB における評価結果

	PTB	MCTB				
		Dialogue	Forum	Law	Literature	Review
Ours	95.17	81.65	83.50	88.28	80.72	79.90
S2S (ours)	93.83	76.62	78.07	82.54	65.34	76.24
S2S-7B (Bai)	95.31	82.92	81.56	84.92	79.19	78.86
S2S-65B (Bai)	95.90	83.72	82.64	85.55	79.73	81.35
Tian ら	96.40	86.01	86.17	91.71	85.27	83.41
Kitaev ら	95.72	86.30	87.04	92.06	86.26	84.34

LLaMa3-8B，Bai らが LLaMa-7B [27] を用いているという点および，我々は単語の品詞タグを利用していないが，Bai らは利用しているという点で異なることに注意されたい．

表より，全く同じ設定で我々の手法と seq2seq を比較すると，すべてのデータセットにおいて提案法が seq2seq を上回っている．特に MCTB における差は顕著である．Bai らの seq2seq と比較しても PTB，Dialogue でやや劣るもののそれ以外では勝っている．Bai らの手法で LLaMa のパラメタサイズを 65B にすると PTB，Dialogue，Review では負けるものの，Forum，Law，Literature では依然勝っている．seq2seq は，入力単語列に対して括弧と非終端記号を挿入した単語列の生成を学習する．つまり，ある種の翻訳を学習する．これに対し，提案法は構文解析の解析動作を学習する．よって，前者は後者よりも入力単語に敏感になり，ドメインが変わると性能劣化が顕著にあらわれると考える．これらより，句構造解析における LLM の活用法としては，単に句構造木を S 式として表現し seq2seq モデルでその変換を学習するより，我々の提案，つまりシフト還元という動作を LLM に学習・予測させることの有効性が示唆される．

次に，Tian らの手法，Kitaev らの手法と比較すると我々の手法は依然劣っており，PTB ではわずかな差であるが，MCTB ではその差は大きい．彼らの手法は BERT や XLNet といった数億（数百 M）のパラメタサイズの小さな事前学習モデルを利用しているのに対し，我々は 80 億（8B）のパラメタサイズの巨大な言語モデルを利用していることを考慮すると，構文解析アルゴリズムそのものの差が影響していると考えられる．ただし，Bai らの seq2seq においてパラメタサイズを大きくすると性能が向上すること，我々の手法が品詞タグを利用していないことを勘案すると，よりパラメタサイズの大きなモデルを利用し，品詞タグを活用することで彼らの手法に近い性能を達成する可能性は十分ある．さらに，我々の手法には手法が単純であるという利点もある．

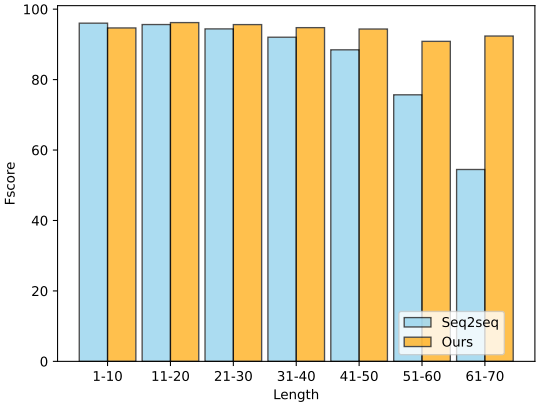

図 3　長さごとの F 値

提案法と seq2seq の違いを考察するため，PTB のテストデータを文長において 10 単語ごとのビンに区切り，それぞれのビンで F 値を計算した．その結果を図 3 に示す．seq2seq は 30 単語を超えると F 値が劣化していき 50 単語以上では F 値 80 を切る．これに対し，提案法はどの長さにおいてもほぼ変わらない F 値である．一般論として，seq2seq による機械翻訳では長い入力に対して翻訳精度が低下することが知られていることから [28, 29]，これは自然な結果と考える．よって，この結果からも LLM に構文解析の動作を学習・予測させることの有効性が示唆される．

6　おわりに

本稿では，句構造解析のため LLM にシフト還元動作を学習・予測させる手法を提案した．同様な手法は Maekawa らによって提案されているが，明示的にスタック，キューをあたえず，それらを，解析位置をあらわす<head>タグを用いて暗黙的に表現する点，解析のすべての状態において，文全体を解析済み S 式と解析前の単語列として参照可能としている点で異なる．提案手法を，LLM を用いた seq2seq モデルと比較評価したところ，PTB での F 値は 95.17 であり，入力文の長さによらず安定して高い F 値を達成した．一方，MCTB での F 値は 80 以上を達成し，異なるドメインのテストデータに対しても高い F 値を得た．これらより，提案法は入力文長，ドメインに対して頑健であることを確認した．

謝辞

本研究は，東京科学大学のスーパーコンピュータ TSUBAME4.0 を利用して実施した.

参考文献

[1] Eric Brill. Automatic grammar induction and parsing free text: a transformation-based approach. In **Proceedings of the Workshop on Human Language Technology**, HLT '93, p. 237–242, 1993.

[2] Michael John Collins. A new statistical parser based on bigram lexical dependencies. In **34th Annual Meeting of the Association for Computational Linguistics**, pp. 184–191, 1996.

[3] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. **Computational Linguistics**, Vol. 19, No. 2, pp. 313–330, 1993.

[4] Michael Collins. Head-driven statistical models for natural language parsing. **Computational Linguistics**, Vol. 29, No. 4, pp. 589–637, 2003.

[5] Adwait Ratnaparkhi. Learning to parse natural language with maximum entropy models. **Machine learning**, Vol. 34, pp. 151–175, 1999.

[6] Jiangming Liu and Yue Zhang. Shift-reduce constituent parsing with neural lookahead features. **Transactions of the Association for Computational Linguistics**, Vol. 5, pp. 45–58, 2017.

[7] David Gaddy, Mitchell Stern, and Dan Klein. What's going on in neural constituency parsers? an analysis. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 999–1010, 2018.

[8] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**, pp. 379–389, 2015.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In **Advances in Neural Information Processing Systems**, Vol. 30, 2017.

[10] Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In **Advances in Neural Information Processing Systems**, Vol. 28, 2015.

[11] Lemao Liu, Muhua Zhu, and Shuming Shi. Improving sequence-to-sequence constituency parsing. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 32, No. 1, 2018.

[12] Xuefeng Bai, Jialong Wu, Yulong Chen, Zhongqing Wang, and Yue Zhang. Constituency parsing using llms, 2023.

[13] Sen Yang, Leyang Cui, Ruoxi Ning, Di Wu, and Yue Zhang. Challenges to open-domain constituency parsing. In **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 112–127, 2022.

[14] Yuanhe Tian, Yan Song, Fei Xia, and Tong Zhang. Improving constituency parsing with span attention. In Trevor Cohn, Yulan He, and Yang Liu, editors, **Findings of the Association for Computational Linguistics: EMNLP 2020**, pp. 1691–1703, 2020.

[15] Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 3499–3505, 2019.

[16] Aru Maekawa, Tsutomu Hirao, Hidetaka Kamigaito, and Manabu Okumura. Can we obtain significant success in RST discourse parsing by using large language models? In Yvette Graham and Matthew Purver, editors, **Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 2803–2815, 2024.

[17] Zizhuo Shen, Yanqiu Shao, and Wei Li. Enhancing discourse dependency parsing with sentence dependency parsing: A unified generative method based on code representation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 12497–12507, 2024.

[18] Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. GoLLIE: Annotation guidelines improve zero-shot information-extraction. In **The Twelfth International Conference on Learning Representations**, 2024.

[19] Kate Thompson, Akshay Chaturvedi, Julie Hunter, and Nicholas Asher. Llamipa: An incremental discourse parser. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 6418–6430, 2024.

[20] Alex Lascarides and Nicholas Asher. **Segmented Discourse Representation Theory: Dynamic Semantics With Discourse Structure**, pp. 87–124. 2007.

[21] W.C. Mann and S.A Thompson. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, USC/ISI, 1987.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 4171–4186, 2019.

[23] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In **Advances in Neural Information Processing Systems**, Vol. 32, 2019.

[24] Aaron Grattafiori, et al. The llama 3 herd of models, 2024.

[25] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: efficient finetuning of quantized llms. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, NIPS '23, 2024.

[26] E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. A procedure for quantitatively comparing the syntactic coverage of English grammars. In **Speech and Natural Language: Proceedings of a Workshop**, 1991.

[27] Hugo Touvron, et al. Llama: Open and efficient foundation language models, 2023.

[28] Masato Neishi and Naoki Yoshinaga. On the relation between position information and sentence length in neural machine translation. In **Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)**, pp. 328–338, 2019.

[29] Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merriënboer, Kyunghyun Cho, and Yoshua Bengio. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In **Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation**, pp. 78–85, 2014.

表 2　ハイパーパラメタ設定

number of training epochs		3
batch size		16
optimizer		paged_adamw_32bit
learning rate	Ours	1e-5
	S2S (ours)	5e-5
learning rate scheduler		Linear warm-up and cosine annealing
warm-up steps		1000
gradient clipping		0.3
lora r		64
lora α		16
lora dropout ratio		0.1
lora target modules		query projection, key projection, value projection, output projection, gate projection, up projection, down projection

A　ハイパーパラメタ

実験に用いたハイパーパラメタを表 2 に示す.