非自己回帰モデルを用いた日本語見出し生成の検討

野崎 樹文 [1,2*]　近藤 雅芳 [1]　内山 達也 [1]
LINE 株式会社 [1]
京都大学 工学部情報学科 [2]
{nozaki.jumon, masayoshi.kondo, tatsuya.uchiyama}@linecorp.com

1 はじめに

スマートフォンの普及と共にインターネットメディアの影響力は益々大きくなりつつある。拡大し続けるインターネットメディア上での効率的なプラットフォーム運営を目的として、掲載記事の自動編集機能のひとつに見出し生成技術がある [1, 2, 3, 4]。見出し生成とは、記事本文を入力し見出しを出力する技術で、要約タスクとして扱われる。

従来、ニューラルネットを用いた見出し生成では、エンコーダ・デコーダモデルを用いて目的の出力系列を逐次的に生成することが一般的であった。このようなモデルは自己回帰モデルと呼ばれ、モデル推論時の時間計算量が出力系列長に比例するため、生成に遅延が生じる欠点がある。

近年、このような問題を解決するアプローチとして、機械翻訳の分野において、出力系列を並列に生成する非自己回帰モデルの研究が活発に進められている [5, 6, 7, 8, 9]。非自己回帰モデルは、出力トークン間の依存関係を捉えにくく現状では性能面で従来のモデルに及ばないものの、出力系列のトークンを並列に出力するため、時間計算量が出力系列長に依存せず、高速な生成を実現できる利点がある。

本研究では、見出し生成技術の高度化を目的として、非自己回帰モデルを用いた日本語見出し生成の検証を行う。具体的には、ニュース記事と見出しのペアから構成されるデータセットに対して、非自己回帰モデルである Conditional Masked Language Model [10] を用いる。また、機械翻訳等の分野において、非自己回帰モデルの性能向上が報告されているいくつかの有効な手法 [11, 12, 13] の適用も検証する。さらに、近年、記事を掲載するメディアのレイアウトや表示デバイスのサイズに合わせて、生成する見出しの出力長に制限を課す技術の研究が積極的に進められている [14, 15] ことから、本研究では非自己回帰モデルの出力長の制御を試みる。具体的には、CMLM が推論時に出力系列長を予め予測することに着目し、その出力系列長を目的の文字数に定めることで正確に出力長を制御する方法を検証する。

* 京都大学の学生 (nozaki.jumon.73s@st.kyoto-u.ac.jp) であり、LINE 株式会社 (京都オフィス) にてエンジニアとして長期アルバイト勤務。

2 事前準備

2.1 問題設定

見出し生成タスクは、S 個のトークンで構成される文 $X = \{x_i\}_{i=1}^{S}$ を入力として与えられた時、T 個のトークン ($T < S$) で構成される尤もらしい見出し $Y = \{y_i\}_{i=1}^{T}$ を出力するモデルを得ることを目的とする。本研究では、日本語のデータセットを用いる。

2.2 非自己回帰モデル

入力文 X が与えられた時に見出し Y を生成する自己回帰モデル $P_{AR}(Y|X)$ は、これまでに生成されたトークン列 $y_{<t}$ の条件付き確率の積として以下のようにモデル化される。

$$P_{AR}(Y \mid X) = \prod_{t=1}^{T} P\left(y_t \mid y_{<t}, X\right) \tag{1}$$

一方、非自己回帰モデル $P_{NAR}(Y|X)$ は、生成トークン間の依存関係が無く、その出力系列の生成確率を入力系列のみに依存する条件付き独立の確率の積として以下のようにモデル化される。

$$P_{NAR}(Y \mid X) = P_L\left(T \mid X\right) \cdot \prod_{t=1}^{T} P\left(y_t \mid X\right) \tag{2}$$

P_L は出力系列長 T を予測する確率関数で、モデルの学習過程で同時に学習される。非自己回帰モデルは、トークンを並列に生成することで生成の高速化が期待できる。一方、トークン間の依存関係が陽にモデル化されないため、自己回帰モデルと同様の学習枠組みでは性能が低くなることが知られている。

2.3 Conditional Masked Langage Model

Ghazvininejad ら [10] は、前述の課題に対するアプローチとして、Conditional Masked Langage Model (CMLM) という非自己回帰モデルを提案した。モデルの学習時は、正解出力系列をランダムにマスクした系列を入力とし、マスクに入るトークンを予測する。一方で、推論時は P_L を用いて予測した系列長のマスクトークンからはじめ、予測尤度の高いトークン とマスクトークンを置き換える操作を予め定められた復号ステップ数だけ繰り返すことによって生成を行う。この時、各復号ステップで埋めるマスクトークン数は、一定になる様にする。

2.4 検証技術

本研究では、非自己回帰モデルを用いた見出し生成に加え、以下の 4 つの手法の検証も行う。

i) 知識蒸留: Sequence-level knowledge distillation [11] は、訓練データの正解ラベルを自己回帰モデルの推論結果で置き換え、そのデータを非自己回帰モデルの訓練データとして用いるという、知識蒸留の手法のひとつである。これにより、multimodality problem [5] が緩和され、学習が安定する効果がある。

ii) 縮約操作: 非自己回帰モデルの生成結果には同一トークンの連続が含まれやすいことが知られており、これに対し Lee ら [12] は連続した同一トークンの重複を排除する手法を提案した。本研究ではこの手法を縮約操作と呼び、検証に用いる。

iii) 語彙分割: 語彙サイズを大きくすることで、各トークンが独立に生成されるという非自己回帰型モデルの仮定の欠点を補う効果があり、精度が向上するという報告がある [13]。本研究では、語彙として文字単位を用いた場合と SentencePiece [16] 用いた場合との精度を比較する。また SentencePiece の語彙サイズを大きくした時の精度の変化を調べる。

iv) 出力系列長の制御: 非自己回帰モデルの出力系列長に制限を課すことを目的として、CMLM の推論時の初期マスクトークンの数を目的の系列長にすることで、陽に出力長を制御する方法を検証する。

3 実験設定

3.1 データセット

本研究では 2 つのデータセットを用いる。ひとつは、インターネットメディアから独自に収集した日本語の Web ニュース記事で構成されたデータセット (以下、**日本語 Web ニュースデータセット**) [1] である。このデータセットは、記事と見出しのペアで構成される。実験の際には訓練データと類似するデータが開発データやテストデータに含まれないようにするため、記事取得時刻順に訓練、開発、テストデータとなる様に分割した。もうひとつは、**livedoor ニュースコーパス**[2] である。このデータセットはテストセットとしてのみ利用する。トークナイズは、文字単位または SentencePiece[3] を用いる。その他のデータセットの詳細は、付録.A に記載する。

3.2 実装の詳細

非自己回帰モデルとして CMLM を用いる。自己回帰モデルには Transformer [17] を用いる。両モデルのエンコーダとデコーダの層数はそれぞれ 1 層ずつ[4] とし、知識蒸留用の教師モデルの Transformer はそれぞれ 3 層ずつとした。推論速度の比較には、バッチサイズを 1 とし、それぞれの実験で 5 回ずつ計測をした平均値を用いた。語彙サイズは、文字単位については訓練データ内で 5 回以上出現する 3392 種とし、SentencePiece については 8000 とした。実装には、fairseq[5] を用いた。その他の実験設定の詳細は、付録.B に示す。

3.3 評価方法

本研究では、以下の 3 つの評価指標を用いる。

ROUGE: 見出しの品質評価を目的に、F1 ベースの ROUGE スコアを用いる。出力系列と正解見出しに対して MeCab [18] よる分かち書きをしてから計算を行う。

Repetition Rate (RR): 非自己回帰モデルの出力系列に対する同一トークンの連続の多さを測る指標として、Repetition Rate (RR) を以下のように定義し用いる。

$$RR = \frac{1}{n}\sum_{i=0}^{n}\frac{r_i}{t_i - 1} \tag{3}$$

t_i は出力系列のトークンの数、r_i は出力系列内の

1) 会員制サイト等を除く、自由閲覧可能なメディアから収集・構築した。非公開のものである。
2) https://www.rondhuit.com/download.html
3) https://github.com/google/sentencepiece
4) データセットサイズが小規模であったため、大きいモデルサイズでは学習が安定しなかった。また、1 層で比較検証するには十分な性能が出たため、本研究では 1 層とした。
5) https://github.com/pytorch/fairseq

	日本語 Web データ				livedoor データ				
モデル	R-1	R-2	R-L	RR	R-1	R-2	R-L	RR	速度
自己回帰モデル									
Transformer (b=1)	35.27	16.88	30.50	0.013	21.81	7.03	18.37	0.020	1.03x
Transformer (b=5)	35.67	18.00	31.34	0.009	21.59	7.60	18.56	0.013	1x
+ Subword	36.79	18.24	32.07	0.008	22.79	8.06	19.34	0.005	1.35x
Transformer (b=20) **	39.08	18.97	33.32	0.008	23.43	7.61	19.44	0.009	0.49x
非自己回帰モデル									
CMLM (k=1)	18.09	3.78	15.94	0.412	10.92	1.01	9.39	0.453	9.76x
CMLM (k=4)	25.10	7.06	21.38	0.163	14.76	2.09	12.25	0.185	5.04x
CMLM (k=8)	29.41	10.17	24.90	0.081	17.48	3.40	14.32	0.090	3.12x
+ 縮約操作	30.98	11.59	28.56	0	18.60	4.12	15.26	0	3.12x
+ 知識蒸留	32.00	13.54	27.81	0.044	19.19	5.36	16.24	0.049	3.10x
+ Subword	33.86	13.47	28.64	0.031	21.36	5.32	17.35	0.061	3.06x
+ 知識蒸留	35.40	14.38	30.42	0.047	20.78	5.10	17.24	0.055	3.08x
+ 縮約操作	36.36	15.18	31.27	0	21.46	5.61	17.85	0	3.08x

表 1: 日本語 Web ニュースデータセットを用いた性能比較。b は推論時のビーム幅、k は CMLM の復号ステップ数を示す。また、** は、CMLM における知識蒸留の教師モデルを示す。**速度**は、 日本語 Web ニュース記事データセットに対するビーム幅 5 の Transformer の生成速度を基準として、各モデルの生成速度の倍率を示している。

トークン単位の bigram の内、同一トークンで構成されるものの数である。

文長の分散値: 文字長制御の効果検証を目的に、高瀬ら [19] に倣い、以下の指標を用いる。

$$var = \frac{1}{n}\sum_{i=0}^{n}|l_i - len|^2 \quad (4)$$

var は、モデルの出力系列長に対する正解系列長との分散を表す指標で、n はデータの数、l_i は生成結果の見出しの長さ、len は正解見出しの長さを表す。

3.4 出力系列長の制御方法

本研究では、モデルの出力系列長の制御方法として、以下の 4 つの方法を検証する。この検証では、トークナイズに文字単位を用いる。

打ち切り: 目的の文字数よりも長い系列を生成した場合に目的の文字数で打ち切り、以降の文字を切り捨てる。

固定長: Rush ら [1] の提案した自己回帰モデルに対する出力長制御手法で、目的の文字数まで終端記号を出力しないことで強制的に目的の文字数までの出力系列を生成する。

制御コード: 人見ら [20] の提案した自己回帰モデルに対する出力長制御手法で、目的の文字数を表す制御コードを入力に加える。本研究では、制御コードとして 6〜69 文字を表す 64 種類を用いる。

固定長初期化: 我々は、CMLM の推論時の初期マストークンの系列長を目的の文字数に定め、陽に出力長を制御する。学習時には、誤差関数から長さ予測に関する項を除いて CMLM の学習を行う。また、人見らの方法と同様に入力に目的系列長を表す制御コードを加えた。

4 実験結果

4.1 自己回帰モデル vs. 非自己回帰モデル

まず、自己回帰モデルの Transformer と非自己回帰モデルの CMLM との性能比較を表 1 に示す。CMLM は、復号ステップ数 k が増えるにつれて ROUGE スコアが大きく改善した。また、縮約操作、知識蒸留、subword の利用といった個々の操作を適用した場合それぞれで ROUGE スコアが改善した。さらに、これら 3 つの操作全てを適用した場合、個々の操作だけの場合よりも大きく ROUGE スコアが改善し、生成速度も約 3 倍程度高速であった。一方で、性能面で依然として自己回帰モデルに及んでいない課題は残る。繰り返し生成の問題については、非自己回帰モデルは自己回帰モデルと比較して RR の値が大きく、同一トークンの連続した生成が多いことが分かる。これに対して、縮約操作を適用することで繰り返し生成の問題が回避され、RR が 0 となるだけでなく、ROUGE スコアも 1 ポイント以上向上した。

表 2 に livedoor ニュースコーパスに対して生成された見出しのサンプルを示す。CMLM の復号ステップ数が大きくなると、文字の繰り返し生成が低減している。さらに、CMLM に 3 つの操作をそれぞれ適用していくと、生成された見出しの可読性が向上していくことが確認できる。また、正解見出しと

正解	ロングラン上映なるか? 『猿の惑星』が好発進
Transformer (b=5)	『猿の惑星:創世記』初日から 4 日間で興収 50 万人突破!
CMLM (k=4)	『猿の記ジェェネネシスス 77 億億億億億円突破破 興収収 30 億億を突破
CMLM (k=8)	『猿星記ジジェネネシス』興収 7 億億億億円超破破 人間で興収 30 億を突破
+ Subword	『猿の記の惑星記記』初登場! 猿衝撃的な興収 730 億円を突破!
+ 知識蒸留	『猿猿の惑星』世界初登場! 4 日間で興収収収 5050 万人突破!
+ 縮約操作	『猿の惑星』世界初登場! 4 日間で興収 50 万人突破!

表 2: livedoor ニュースコーパスを用いて生成された見出しのサンプル。サンプルの正解見出しと各モデルにより生成された見出しをそれぞれ示す。トークナイズには、文字単位を用いた。

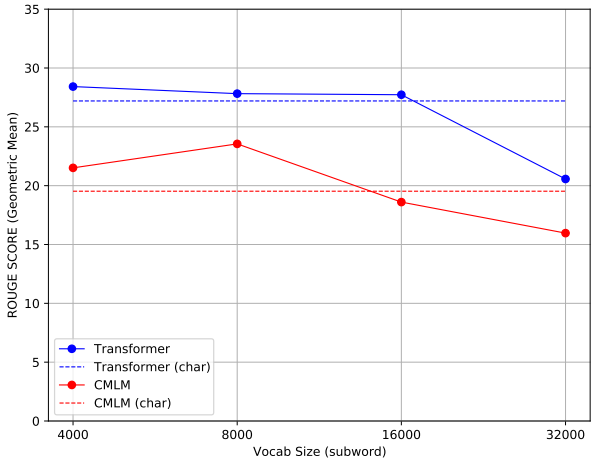

図 1: subword の語彙サイズを変更した場合の CMLM と Transformer のそれぞれの精度を示す (実線)。縦軸は 3 つの ROUGE 値 (R-1, R-2, R-L) の幾何平均値を、横軸は語彙サイズを示す。また、破線はトークナイズに文字単位 (size:3392) を適用した場合の精度を示している。データセットは、日本語 Web ニュースデータを用いた。本実験の詳細なスコアは、付録.C に記載。

各モデルの生成見出しの差異が大きい要因として、学習時とテスト時で異なるデータセットを用いたことが考えられる。

4.2 語彙と性能の関係

次に、図 1 に語彙分割の方法と性能との関係を示す。subword の語彙数が、自己回帰モデルでは 4,000 の時に、非自己回帰モデルでは 8,000 の時に、それぞれ最も性能が高かった。また、トークナイズに文字単位と subword を用いた場合で比較すると、非自己回帰モデルの方が自己回帰モデルよりも性能の向上幅が大きかった。一方で、自己回帰モデルと非自己回帰モデルで共に、subword の語彙サイズが 16,000 以上になると性能が低下した。これは、今回用いたデータセットのサイズが比較的小さく、大きな語彙サイズによる学習が安定しなかったことが要因であると考えられる。

モデル	R-1	R-2	R-L	var
Transformer (b=5)	35.67	18.00	31.34	226.3
+ 打ち切り	35.21	17.77	30.89	122.4
+ 固定長 [1]	35.02	17.11	30.03	0
+ 制御コード [20]	35.94	17.77	31.11	1.1
CMLM (k=8)	29.41	10.17	24.90	105.3
+ 打ち切り	28.77	9.91	24.40	75.27
+ 固定長初期化 (proposed)	32.00	12.16	27.13	0

表 3: 出力系列長の制御を適用した場合の性能を示す。データセットは、日本語 Web ニュースデータセットを用いた。トークナイズは、文字単位を用いた。

4.3 出力系列長の制御

最後に、表 3 に、各モデルに出力系列長の制御の手法を適用した場合の性能を示す。自己回帰モデルでは、全ての制御方法において正解系列長との分散の値は小さくなる一方、性能の低下が確認された。それに対して、非自己回帰モデルでは、目的文字数以降の文字を打ち切る方法では性能が低下したが、初期化トークンを目的文字数に固定する方法では正確に長さ制御を行いつつ性能が向上した。これは、目的の文字数を正確に出力していることに加え、損失関数から長さ予測に関する項が取り除かれることで学習が安定したことが理由として考えられる。

5 おわりに

本研究では、非自己回帰モデルを用いた見出し生成を行い、自己回帰モデルとの性能比較を行った。また、機械翻訳等で用いられている、非自己回帰モデルの性能向上を目的としたいくつかの手法について、日本語見出し生成タスクでも同様の有効性を確かめた。さらに、非自己回帰モデルによる生成時に、生成長を予め与える簡易な方法で性能を向上させながら正確に出力長を制御できることを示した。一方で、未だ自己回帰モデルの生成する見出しとの品質差は大きい。今後は、非自己回帰モデルの精度をより高めていくために大規模な事前学習済み言語モデル [21, 22, 23, 24] などの利用を検討予定である。

参考文献

[1]Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, 2015.

[2]Kazuma Murao, Ken Kobayashi, Hayato Kobayashi, Taichi Yatsuka, Takeshi Masuyama, Tatsuru Higurashi, and Yoshimune Tabuchi. A case study on neural headline generation for editing support. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pp. 73–82, 2019.

[3]人見雄太, 田口雄哉, 田森秀明, 岡崎直観, 乾健太郎. 小規模リソースにおける生成型要約のためのスタイル転移. 言語処理学会第 26 回年次大会, 2020.

[4]小林健, 小林隼人, 村尾一真, 増山毅司. 複数エンコーダを用いたヤフートピックス見出し候補生成. 言語処理学会第 26 回年次大会, 2018.

[5]Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018.

[6]Jiatao Gu, Changhan Wang, and Junbo Zhao. Levenshtein transformer. In *Advances in Neural Information Processing Systems*, pp. 11181–11191, 2019.

[7]中村朝陽, 鶴岡慶雅. 非自己回帰的な生成と事前学習を用いた機械翻訳への試み. 言語処理学会第 26 回年次大会, 2020.

[8]安井豪, 鶴岡慶雅, 永田昌明. モデルに基づくマスキングを行う非自己回帰的ニューラル機械翻訳. 言語処理学会第 26 回年次大会, 2020.

[9]朱中元, Jason Lee, Kyunghyun Cho, 中山英樹. 潜在変数の精緻化による非自己回帰型ニューラル機械翻訳. 言語処理学会第 26 回年次大会, 2020.

[10]Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6114–6123, 2019.

[11]Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016.

[12]Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1173–1182, 2018.

[13]Hirofumi Inaguma, Yosuke Higuchi, Kevin Duh, Tatsuya Kawahara, and Shinji Watanabe. Orthros: Non-autoregressive end-to-end speech translation with dual-decoder. *arXiv preprint arXiv:2010.13047*, 2020.

[14]Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pp. 45–54, 2018.

[15]Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1328–1338, 2016.

[16]Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, 2018.

[17]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[18]Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 230–237, 2004.

[19]Sho Takase and Naoaki Okazaki. Positional encoding to control output sequence length. 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. 3999–4004, 2019.

[20]Yuta Hitomi, Yuya Taguchi, Hideaki Tamori, Ko Kikuta, Jiro Nishitoba, Naoaki Okazaki, Kentaro Inui, and Manabu Okumura. A large-scale multi-length headline corpus for analyzing length-constrained headline generation model evaluation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 333–343, 2019.

[21]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, 2019.

[22]Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020.

[23]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 preprint arXiv:1907.11692*, 2019.

[24]Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 2020.

A　データセットの詳細

日本語 Web ニュース記事データセット: データサイズについては、それぞれ、訓練データが 5 万以上、開発が 4,908、テストが 4,887 サンプルとなっている。

livedoor ニュースコーパス[6]: サンプル数は 6,928 となっている。本研究には、テストセットにのみ用いている。

B　実験設定の詳細

モデル訓練時の設定は、dropout 率を 0.3、学習率を 0.0005、warm-up steps を 10000、weight decay を 0.01 と設定した。学習ステップ数は最大で 50 万とした。バッチサイズは、バッチ内トークン数が最大 8000 となるように動的に構成するようにした。モデル学習時の入出力については、最大入力長と最大出力長をそれぞれ 128 語、69 語として学習を行う。一方で、推論時には最大出力長を 200 語として生成を行った。その他のパラメータ設定は、Vaswani ら [17] の Transformer (base model) と同様のものを用いている。

実験環境には、NVIDIA Telsa V100 GPU を 1 枚利用した。また、CPU は Intel Xeon Silver 4114 CPU (2.20GHz) を 2 個使用し、RAM は 24 GB とした。

C　語彙と性能の関係の詳細

モデル	R-1	R-2	R-L
Transformer char	35.67	18.0	31.34
Transformer 4k	37.31	18.88	32.59
Transformer 8k	36.79	18.25	32.07
Transformer 16k	37.25	17.88	32.01
Transformer 32k	30.67	10.84	26.17
CMLM char	29.41	10.17	24.9
CMLM 4k	31.48	11.81	26.78
CMLM 8k	33.86	13.47	28.64
CMLM 16k	29.17	9.04	24.42
CMLM 32k	25.21	7.5	21.53

表 4: 語彙サイズを変更した場合の CMLM と Transformer の各 ROUGE スコアを示す。char は文字単位 (size:3392) を示す。また、表記の 4k, 8k, 16k, 32k は、それぞれ subword の語彙サイズを示す。

6） https://www.rondhuit.com/download.html