日本語 GPT を用いたトークナイザの影響の調査

井上 誠一 [1 2] Nguyen Tung [1] 中町 礼文 [1] 李 聖哲 [1] 佐藤 敏紀 [1]
[1]LINE 株式会社 [2] 東京都立大学
{seiichi.inoue, tung.nguyen, akifumi.nakamachi,
shengzhe.li, toshinori.sato}@linecorp.co.jp

概要

本研究では，大規模言語モデルである日本語 GPT を用いて，Byte-level Byte Pair Encoding トークナイザの構築方法や語彙サイズの違いによる言語モデルの性能を比較し分析した．具体的には，日本語テキストを対象としたトークナイザ構築において，トークナイザの構築に用いるテキストの事前分割，トークナイズ時の分かち書きの有無，語彙サイズという観点で調査を行った．

1 はじめに

近年，GPT-3 [1] をはじめとした大規模言語モデルが注目を集めており，自然言語処理のさまざまなタスクの性能を向上させている．言語モデルの構築において，モデル化するテキストの単位を検討することは重要であるが，モデルのアーキテクチャや言語，またタスクによって適切なトークナイザは異なる [2]．例えば，Alyafeai ら [3] は，アラビア語において，複数の異なるトークナイズ手法を用いて感情分析と文書分類を行っており，ほとんどのタスクにおいて，単語レベルのトークナイザを小さな語彙サイズで学習させたものが最良の結果となることを示した．また，Pan ら [4] は，トルコ語とウルグアイ語の機械翻訳において，形態素解析を行なった後に Byte Pair Encoding (BPE) [5] の学習を行うことで，通常の BPE を使用する場合に比べて性能が向上することを示した．しかし，これらは GRU [6] や Transformer [7] を対象としており，GPT におけるトークナイザの比較，分析は行われていない．また，GPT では，トークナイザに Byte-level Byte Pair Encoding (Byte-level BPE) [8] が用いられるが，現時点では対象の言語が英語や韓国語 [1, 9] に限られており，日本語を中心としたコーパスに対する Byte-level BPE の分析は行われていない．

そこで，本研究では，条件の異なるトークナイザを用いて，日本語 GPT の事前学習と含意関係認識タスクを用いた転移学習を行い[1)]，トークナイザの違いが言語モデルの事前学習や下流タスクに与える影響を示し，分析する．

以下に本研究の貢献を示す:

- Byte-level BPE トークナイザについて，トークナイザの構築に用いるテキストの事前分割，トークナイズ時の分かち書きの有無，語彙サイズという観点で比較を行い，分析した．
- 日本語 GPT を用いた含意関係認識タスクにおけるトークナイザの違いによる影響を比較し，分析した．

2 関連研究

2.1 Generative Pre-training of a Language Model (GPT)

GPT [1, 10, 11] は，Transformer decoder をベースとした，大規模なコーパスを用いて教師なしで学習される自己回帰型言語モデルである．具体的には，コーパス $\mathrm{X} = (x_1, x_2, \ldots, x_N)$ の可変長のトークン列を $x_n = (s_1, s_2, \ldots, s_M)$ としたとき，以下の目的関数を最大化することで学習される:

$$p(x_n) = \prod_{i=1}^{M} P(s_i \mid s_{i-k}, \ldots, s_{i-1}). \quad (1)$$

ただし，k は文脈窓幅であり，条件付き確率 P はニューラルネットワークを用いてモデル化される．

2.2 Byte-level Byte Pair Encoding

GPT では，トークナイザとして，頻度に基づいた単語分割を学習する手法の一つである Byte-level BPE が用いられている．BPE が文字レベルでのトークンを学習するのに対し，Byte-level BPE は Byte レベルでトークンを学習しており，基本的には BPE と

1) GPT の構築には Megatron-LM を用いた: https://github.com/NVIDIA/Megatron-LM.

同じアルゴリズムで構築される．

Kim ら [9] は，韓国語を中心としたコーパスを用いて GPT を構築した．トークナイズにおいて，韓国語は英語とは異なる言語特性をもつため，生のテキストを用いてトークナイザの構築を行うのではなく，形態素解析器を用いてコーパスを事前に形態素に分割してから Byte-level BPE を学習している．しかし，日本語は単語と単語の間にスペースが含まれないなど，韓国語とも言語的な特徴が異なっているため，日本語において同様の処理を行うことが適切であるかは自明でない．

2.3 トークナイザ分析

日本語トークナイザ構築における辞書選択 菊地ら [12] は，日本語 BERT を対象として，トークナイザの入力時において文章を形態素に分割する際に用いる辞書の違いによる性能を，分類タスクを用いて比較した．しかし，菊池らは，事前学習とファインチューニングでトークナイザを統一していないため，トークナイザ選択による影響を正確に調査できていない．また，これはトークナイザに SentencePiece [13] を用いた BERT に限定された研究であるため，GPT における Byte-level BPE において同様の結果となるかは定かではない．

語彙サイズの選択 Gowda ら [14] は，機械翻訳における適切な語彙サイズに関する分析を行っており，BPE において語彙サイズを大きくすることが性能の向上につながることを示した．また，Xu ら [15] も，機械翻訳において，語彙サイズを変化させた時の BLEU [16] のスコアの変化量（Marginal Utility of Vocabularization: MUV）について最適輸送を用いて最適化する手法を提案した．しかし，これらは日本語を対象とはしておらず，日本語コーパスにおけるトークナイザの語彙サイズについては分析されていない．

2.4 含意関係認識

含意関係認識とは，以下に示すようなテキストと仮説を与え，テキストと仮説の間に含意，矛盾，中立のどの関係があるかの判別をシステムで行うタスクである．

テキスト 鉢植えの植物のあるテーブルの前で，笑顔の男性の隣に座っている女性が笑っている．
仮説 女性は笑っている．
推論判定 含意

表 1 前処理後の JSNLI データセットの統計量．

データ	サンプル数	含意	矛盾	中立
学習	47,970	15,974	15,965	16,031
開発	5,330	1,689	1,844	1,797
テスト	3,916	1,432	1,156	1,328

3 実験

3.1 データセット

事前学習 GPT の事前学習には，Wikipedia，ニュースサイト，ブログサイト，新聞等の日本語を中心としたテキストから構築されたコーパスを用いた．前処理として，重複する文書の削除，ランダムサンプリングを行い，最終的に学習に使用したコーパスのサイズは 10GB となった．

評価タスク 本研究では，GPT の性能比較のための評価タスクとして日本語 SNLI （JSNLI） データセット [17] を用いた．JSNLI は，含意関係認識のデータセットである SNLI を日本語に翻訳したものであり，SNLI に機械翻訳を適用した後，評価データにクラウドソーシングによるフィルタリング，学習データに計算機による自動フィルタリングを施すことで構築されている．本研究では，公開されているデータセットのうち，フィルタリング後の学習データを用い，それに対し学習コストの問題から，さらに 1/10 のランダムサンプリングを行い，それを 9:1 で学習データと開発データに分割した．評価データについては，公開されているものをそのまま用いた．表 1 にデータセットの統計量を示す．

3.2 事前学習

GPT の事前学習において，モデルの Layer 数を 12，Hidden dimension を 512，Attention Head の数を 8 とした．また，学習率は 1e-4，バッチサイズは 56 [2] とし，iteration 数は 1,000,000 とした．

3.3 転移学習

GPT で評価タスクを解く際は，転移学習やファインチューニングによるモデルの調整を行わず，推論時にタスクに関する説明と少量のデモンストレーションを与える few-shot 設定を用いるのが一般的である．しかし，本研究では，計算コストの問題から GPT のパラメータ数を小さくしており，few-shot 設

2） 本研究では，計算機のメモリの都合上この値を採用した．

定では十分な性能が出せないと予測できるため，評価セットを用いた転移学習を行う．含意関係認識タスクに向けた転移学習の際は，Zhao ら [18] に従い，テキストと仮説に続く推論判定を文字列で予測させた．入力に続く文字列を予測させて評価する際は，予測の先頭のトークンを用いて正誤を判定するのが一般的である [19]．そのため本研究では，正誤判定に用いる予測ラベルを 1 トークンで表現できるように，含意 → “正”，矛盾 → “誤”，中立 → “不可” と正解ラベルを設定した．転移学習において，モデルのアーキテクチャは事前学習と同様にし，学習率を事前学習時より小さい値の 5e-5，バッチサイズを 32 とした．また，iteration 数は 15,000 とし，以下では全て最後の iteration におけるモデルのパラメータを用いて評価を行った．

3.4 実験設定

本研究では，GPT のトークナイザとして Byte-level BPE を用いる．トークナイザの比較条件を以下の 3 点とした:

- トークナイザ構築時のテキストの事前分割の有無と辞書選択
- トークナイズ時の分かち書きの有無
- トークナイザ構築時の語彙サイズの選択

そして，次に示すように 2 ステップに分けて実験を行なう．

辞書選択と事前分割に関する予備実験　まず，トークナイザ構築において，予備実験では語彙サイズを 50,257 [3] で固定とする．また，トークナイザ構築に用いるテキストは，unidic [4] / ipadic [5] を用いて事前分割されたものに加えて事前分割を行わないものの 3 種類とする．これらのトークナイザを用いて GPT の事前学習，転移学習を行い，スコアを元に比較を行う．ただし，上記のトークナイザのうち，事前分割を行ったテキストを用いて構築されたものは，トークナイズ時に分かち書きを行う場合と行わない場合での比較も行う．予備実験の詳細を表 2 に示した．

語彙サイズの実験　語彙サイズの実験では，事前分割の有無と辞書選択，トークナイズ時の分かち書きの有無を，下流タスクの性能が予備実験において最良だったもので固定とする．語彙サイズについては，25,257 から 125,257 まで 25,000 ずつ変化させ，予備実験と同様に GPT の事前学習，転移学習を行い，スコアを元に比較を行う．

3） GPT では，語彙サイズを基本語彙として 256 トークン，文末トークンとして 1 トークン，それらにマージ数を加えたものとしている．
4） https://ccd.ninjal.ac.jp/unidic/
5） https://taku910.github.io/mecab/

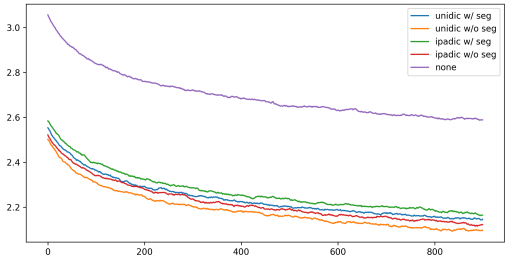

図 1　辞書選択と事前分割に関する予備実験における各トークナイザの事前学習における validation loss の推移

3.5 予備実験の結果

まず，辞書選択と事前分割に関する予備実験における各トークナイザの統計量を表 2 に示す．事前分割したテキストを用いて構築されたトークナイザに対しては，トークナイズ時の分かち書きを行った場合と行わなかった場合のトークン化された文章長をそれぞれ別途記載した．トークン長については，事前分割の有無，辞書の違いによって大きな差がないことがわかる．他と比べて，none のトークン長の標準偏差と比べて大きくなっているが，これは，辞書を用いて形態素に事前分割されたテキストを用いてトークナイザを構築する場合，形態素を超えてマージが行われないが，生のテキストからマージを学習する none は比較的長いトークンができやすいからであると推測できる．文章長については，上述の理由から none が一番短くなっている他，トークナイズ時の分かち書きを行った方が僅かに文章長が短くなることが確認された．

これらのトークナイザと条件を用いて事前学習を行った際の loss の推移を図 1 に，転移学習を行なった結果を表 2 に示した．事前学習の結果からは以下の条件:

- トークナイザ構築に用いるテキストの事前分割を行う
- 事前分割には unidic を用いる
- トークナイズ時の分かち書きは行わない

で性能が高いことがわかる．転移学習では，評価タスクのスコアは事前学習の結果の傾向と異なり，トークナイズ時の分かち書きを行った方が性能が高

表 2　辞書選択と事前分割に関する予備実験における各トークナイザの統計量と転移学習の結果

	事前分割	辞書	分かち書き	token len (std.)	sentence len (std.)	acc (%)
unidic w/ seg	あり	unidic	あり	3.27 (1.42)	377.87 (815.81)	**84.76**
unidic w/o seg			なし		395.34 (871.35)	83.68
ipadic w/ seg		ipadic	あり	3.30 (1.46)	352.46 (765.53)	**84.76**
ipadic w/o seg			なし		382.58 (844.64)	83.55
none	なし	—	—	3.28 (2.01)	283.73 (616.20)	83.88

表 3　語彙サイズの実験における各トークナイザの統計量と転移学習の結果

語彙サイズ	token len (std.)	sentence len (std.)	acc (%)
25257	2.95 (1.29)	394.87 (857.24)	84.32
50257	3.27 (1.42)	380.18 (823.84)	84.75
75257	3.45 (1.52)	374.40 (810.67)	84.35
100257	3.59 (1.59)	371.26 (803.59)	**85.42**
125257	3.69 (1.66)	369.33 (799.20)	83.47

い結果となった．

以上の結果を踏まえて，語彙サイズの実験には下流タスクで最もスコアが高かった unidic w/ seg の条件を用いる．

3.6　語彙サイズの実験の結果

語彙サイズの実験における各トークナイザの統計量を表 3 に示す．語彙サイズが大きくなるに従って，トークン長の平均が長くなり，トークン化された文章長が短くなっていることがわかる．これはマージが増えることでトークン化の際の語彙のマッチ率が上がるためと推測できる．

事前学習における loss の推移を図 2 に，転移学習の結果を表 3 に示す．事前学習においては，語彙サイズが小さい，つまり文章長が長くなるトークナイザを用いた方が性能が高い傾向がみられた．転移学習での結果は，予備実験での結果の際と同じように事前学習時と異なっており，loss が低くなる小さな語彙サイズで構築されたトークナイザではなく，語彙サイズが比較的大きめな 100,257 のトークナイザを用いた際にスコアが最も高くなっている．

4　議論

予備実験と語彙サイズの実験を通して，トークナイザの統計量と性能の関係性を観察すると，事前学習においては「文章長が長い方が性能が高くなる」という仮説が得られる．これは，文章長が長くなると，モデルの計算量は増加するため [6]，スケーリングの法則 [20] に類似した現象である可能性も考えられる．予備実験において，転移学習の結果とは異なり，事前学習においては，分かち書きを行わうと性能が悪くなる傾向がみられた．これは，上述の仮説に基づくと，分かち書きを行うことにより，入力がトークナイザ構築に用いたテキストと同じになるため，トークナイズ時の語彙のマッチ率が上がり，平均的に文章長が短くなることで事前学習の性能が悪くなっていると考えられる．語彙サイズの実験においても，同様に結果を解釈することができる．一方で，予備実験，語彙サイズの実験のどちらにおいても，転移学習時は必ずしも上述の仮説通りの結果とはなっていない．これは，下流タスクでは，トークンの持つ性質として，仮説のような機械学習的にメリットのある性質よりも，言語学的な性質の方が重要な場合があるからではないかと考える．[7]

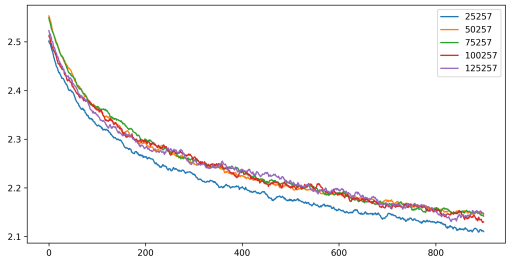

図 2　語彙サイズの実験における各トークナイザの事前学習における validation loss の推移

5　まとめと今後の展望

本研究では，日本語 GPT におけるトークナイザの影響を調査し，様々な条件における性能比較と分析を行った．今後は，4 節での考察を元に，機械学習的な性質と言語学的な性質の関係性を明らかにするため，他の下流タスクを用いた評価を含めたさらなる調査を行っていきたい．

6）　特に，attention 部分の計算が 2 乗のオーダーで大きくなる．

7）　仮説の極端な例として，「語彙が基本語彙の 256 トークンしかないトークナイザ」を考えると，ほとんどの単語が最小単位で表現され，文章長が長くなるが，事前学習時の性能が仮に良かったとしても，トークナイズされた文章が言語学的に意味を持っているとは言い難い．

参考文献

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. Language Models Are Few-Shot Learners. Advances in Neural Information Processing Systems, vol. 33, 1877–1901.

[2] Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, et al. 2021. Between Words and Characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. arXiv [cs.CL]. arXiv. http://arxiv.org/abs/2112.10508.

[3] Zaid Alyafeai, Maged S. Al-shaibani, Mustafa Ghaleb, and Irfan Ahmad. 2021. Evaluating Various Tokenizers for Arabic Text Classification. arXiv [cs.CL]. arXiv. http://arxiv.org/abs/2106.07540.

[4] Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. 2020. Morphological Word Segmentation on Agglutinative Languages for Neural Machine Translation. arXiv [cs.CL]. arXiv. http://arxiv.org/abs/2001.01589.

[5] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, 1715–25.

[6] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Presented in NIPS 2014 Deep Learning and Representation Learning Workshop.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Ł. Ukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In Advances in Neural Information Processing Systems, vol. 30, 5998-6008.

[8] Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural Machine Translation with Byte-Level Subwords. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 9154-9160.

[9] Boseop Kim, Hyoungseok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Dong Hyeon Jeon, Sunghyun Park, et al. 2021. What Changes Can Large-Scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-Scale Korean Generative Pretrained Transformers. arXiv [cs.CL]. arXiv. http://arxiv.org/abs/2109.04650.

[10] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training.

[11] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and Others. 2019. Language Models Are Unsupervised Multitask Learners. OpenAI Blog 1 (8): 9.

[12] 築地 俊平, 新納 浩幸. Tokenizer の違いによる日本語 BERT モデルの性能評価. 2021. 言語処理学会 第 27 回年次大会.

[13] Taku Kudo, and John Richardson. 2018. 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, 66–71.

[14] Thamme Gowda, and Jonathan May. 2020. Finding the Optimal Vocabulary Size for Neural Machine Translation. In Findings of the Association for Computational Linguistics: EMNLP 2020, 3955–64.

[15] Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary Learning via Optimal Transport for Neural Machine Translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, vol. 1, 7361–73.

[16] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 311–18.

[17] 吉越 卓見, 河原 大輔, 黒橋 禎夫. 2020. 機械翻訳を用いた自然言語推論データセットの多言語化. 第 244 回自然言語処理研究会.

[18] Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before Use: Improving Few-Shot Performance of Language Models. arXiv [cs.CL]. arXiv. http://arxiv.org/abs/2102.09690.

[19] Neel Alex, Eli Lifland, Lewis Tunstall, Abhishek Thakur, Pegah Maham, C. Jess Riedel, Emmie Hine, et al. 2021. RAFT: A Real-World Few-Shot Text Classification Benchmark. arXiv [cs.CL]. arXiv. http://arxiv.org/abs/2109.14076.

[20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. arXiv [cs.LG]. arXiv. http://arxiv.org/abs/2001.08361.