単語単位のトークン分割を用いた LLM の性能調査

清水 誉大 [1]　王 天奇 [1]　鈴木 潤 [1,2,3]
[1] 東北大学　[2] 理化学研究所　[3] 国立情報学研究所 LLMC
is-failab-research@grp.tohoku.ac.jp

概要

一般公開されている学習済みの大規模言語モデル（LLM）では，サブワードに基づくトークン分割を採用しているものが多い．それに対し，LLM 以前によく用いられていた単語単位のトークン分割を LLM に採用した場合の効果を検証する．ただし，未知語対策としてバイトフォールバック（byte fallback）を採用し，バイト列に変換することで未知トークンを防ぐ方法を用いる．単語単位のトークン分割を用いて事前学習したモデルと，標準的なサブワード単位のトークン分割を用いて事前学習したモデルを用意し，複数の質問応答タスクで評価した．それぞれの結果を分析し，トークン分割の違いが LLM に与える影響を調査した．

1　はじめに

大規模言語モデル（LLM）の入力となる文章は，通常，事前に定義されたトークン分割器によってトークン列に変換される．現在の LLM の多くは，サブワード（subword）[1, 2, 3, 4] と呼ばれる単位に分割するトークン分割器を採用している．サブワード単位が用いられる主な理由は，語彙数を一定に抑えたまま未知トークンの出現を抑制できる点にある．低頻度語をサブワードの組み合わせで表現することで，例えば英語では未知トークンの発生をほぼ防げる．しかし，多言語入力を想定すると，文字体系の多様性から，語彙に含まれない文字が頻出し，その結果，トークン分割後のトークン列に未知トークンが発生する．この状況に対処するため，バイトフォールバックという考え方が考案された [5]．バイトフォールバックでは，例えば語彙に含まれない文字が出現した場合，それらをすべてバイト単位のトークン列として扱う．これにより，トークン化効率などを無視すれば，理論上，語彙に含まれない文字に起因する未知トークン問題を解消できる．表 1 に，単語単位とサブワード単位のトークン分割の違いの例を示す．

表 1　トークン分割器の違いに基づく挙動の違い（バイトフォールバックを前提）．ここでは仮に *synaptic* が辞書に含まれない場合を示す．

例文	*Neurotransmitter release regulates synaptic communication.*
単語単位	`Neurotransmitter / release / regulates / <0x73> / <0x79> / <0x6E> / <0x61> / <0x70> / <0x74> / <0x69> / <0x63> / communication / .`
サブワード単位	`Neuro / trans / m / itter / release / regulates / syn / ap / tic / communication / .`

言語における「単語[1)]」は，文章理解における重要な処理単位であると考えられている [6, 7]．そのため，LLM においても単語単位のトークン列として処理する方が文章の意味をより把握しやすくなり，様々なタスクで高い性能を示す可能性があると想定される．そこで本研究では，この仮説を検証する．具体的には，バイトフォールバックを共通の未知トークン対策の前提技術とした上で，従来用いられていた単語単位のトークン分割と，近年一般的なサブワード単位のトークン分割を LLM のトークン分割器として用いてモデルを学習し，それぞれのモデルの性能差を分析することで，単語単位のトークン分割が LLM での利用において効果的であるかを検証する．

2　関連研究

サブワードが導入される以前のニューラル機械翻訳では，トークン分割は主に単語単位で行われてきた [8, 9, 10]．その際，語彙に含まれない単語は，未知語を表す特殊トークン（例：<UNK>）に置き換えられていた．そのため，語彙数の増大と未知トークン率の低減の間のトレードオフが課題となっていた．

これに対し，サブワードによるトークン分割では，文章を文字より大きく，単語より小さい単位

1)　厳密には「単語」の定義が明確ではないが，読みやすさを優先して本稿では「単語」という用語を用いる．

に分割し[2]，それらの組み合わせとして語を表現する [2]．この枠組みにより，高頻度語は 1 つのトークンとして保持されやすく，低頻度語や未知語は複数のサブワード列として表現されるようになる．

SentencePiece [5] は，言語に依存しないトークン分割を目的として提案されたサブワード分割手法およびツールである．事前の単語分割や空白情報に依存せず，生テキストから直接サブワード語彙を構築できる点に特徴がある．この性質により，空白を用いない言語や，単語境界が明確でないテキストに対しても一貫したトークン分割を提供する．

また，SentencePiece はバイトレベル表現を導入しており，学習語彙に含まれない文字列に対しても，文字列をバイト列に分解することでトークン分割できる．この機能をバイトフォールバックと呼ぶ．

サブワード単位でのトークン分割において，考案されたバイトフォールバックであるが，以前の単語単位のトークン分割の際にも概念的に競合しないため，組み合わせて利用可能である．本稿では，この性質を利用し，バイトフォールバック付きの単語単位のトークン分割を利用する．

3 トークン分割器の作成

3.1 設計方針

単語単位のトークン分割（Word-level Tokenization）は，サブワード単位の手法と比較して未知語への対応能力は劣るものの，単語の意味を直接的にモデルの入力として扱えるという利点がある． 本研究では，サブワード分割を行わず，単語をそのままトークン化するトークン分割器を構築する．具体的には，1 トークンとして表現する単語群からなる辞書を構築し，一般語を管理する．一方，辞書に含まれない固有名詞や低頻度語については，バイトフォールバックを用いて表現する手法を採用する．

3.2 辞書構築

単語単位のトークン分割器の語彙として用いる目的で単語辞書を作成した．基礎的な語彙として，一般語を高い品質で包含する Oxford 3000 [3] を採用した．バイトフォールバックの 256 語を含めて合計 3,231 語からなる初期辞書を構築した．

初期辞書に単語を追加して，表現可能な語彙を拡張した．語彙拡張にあたっては，smollm-corpus [11][4] に含まれる fineweb-edu-dedup サブセットから文書を抽出し，そこに出現する単語を頻度順に整理した．頻度推定用コーパスから抽出した単語を，出現頻度の高い順に辞書へ追加することで，語彙を段階的に拡張した．

3.3 被覆率計測

単語辞書の語彙拡張を行うにあたり，適切な語彙数を決定する必要がある．自然言語においては，新たな単語が継続的に生成されるため，理論的には語彙集合は無限である．一方で，一般語は比較的有限な集合として捉えられるのに対し，固有名詞や専門語は際限なく増加するという性質を持つ．このことから，一般語や高頻度語を優先的に辞書へ追加することで，比較的少ない語彙数で多様な表現を可能にできると考えられる．このような特性を踏まえると，語彙数を増加させた際に，コーパス中に出現する単語をどの程度カバーできるかを定量的に評価することが重要である．

本研究では，語彙数を段階的に増加させた際のコーパスに対する被覆率を評価するため，被覆率を指標として計測した．被覆率は，辞書に含まれる単語のみを用いて，評価コーパス中のテキストをどの程度表現できるかを表す指標である． 評価には，smollm-corpus の fineweb-edu [12, 13] サブセットを用い，頻度推定用とは独立に評価用コーパスを用意した．初期辞書として Oxford 3000 を用い，語彙数を段階的に拡張しながら，最大 100,000 語まで被覆率を計測した．

図 1 は，語彙数を段階的に増加させた際の被覆率の推移を示している．語彙数の増加に伴い，被覆率は単調に上昇する傾向が確認できる一方で，その増加率は一定ではないことが分かる．

3.4 被覆率についての考察

図 1 より，語彙数が小さい領域では，被覆率は急速に増加しており，少数の単語を追加するだけで，評価コーパス中の多くの語を新たにカバーできていることが分かる．この段階では，一般語や高頻度語が優先的に語彙へ取り込まれていることを反映していると考えられる．

2） 分かち書きされない言語では，単語より大きな単位になる場合もあるが，ここでは単純化して説明している．

3） https://www.oxfordlearnersdictionaries.com/external/pdf/wordlists/oxford-3000-5000/The_Oxford_3000.pdf

4） https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus

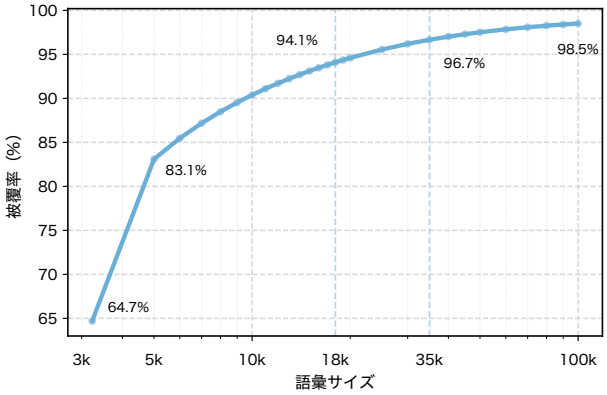

図 1　語彙数と被覆率の関係.

一方で，語彙数が大きくなるにつれて，被覆率の増加は次第に緩やかになる．特に高い被覆率を達成した後では，語彙を大幅に追加しても，被覆率の改善は限定的である．この領域では，主に低頻度語や固有名詞が追加されており，それらがコーパス全体に占める割合が小さいことを反映している．

以上の結果は，語彙数の増加に伴い，辞書へ単語を追加することによる被覆率の改善効果が逓減していくことを示している．このことは，高頻度な一般語を語彙として管理することには大きな効果がある一方で，低頻度語や固有名詞を 1 トークンとして扱うことの寄与は限定的であることを意味する．

そこで本研究では，初期辞書に対して出現頻度順に単語を追加し，被覆率の増分が大きい領域に属する語を中心に 1 トークンとして表現する単語辞書を構築し，それ以外の語についてはバイトフォールバックによる表現に委ねる方針を採用する．

4　実験設定

本稿の実験は，バイトフォールバックを導入した単語単位のトークン分割器とサブワード単位のトークン分割器だけを変更して構築した事前学習済みモデルの性能を評価し，それぞれのトークン分割器の効果を比較検証することを目的とする．

4.1　比較対象

MP モデル　前節で説明した辞書を語彙とした単語単位のトークン分割器を用いる．語彙に含まれない単語は全てバイトフォールバックによりバイト列として扱われる．

SPM モデル　昨今の LLM にてよく用いられるサブワード単位のトークン分割器を用いる．SentencePiece を使用して構築する．

4.2　事前学習用モデルの設定

モデルアーキテクチャ　ベースのモデルアーキテクチャーは SmolLM を採用した．また，モデルサイズ（パラメータ数）は 135M を選択した．

学習データ　事前学習用の学習データとしては，データの質が比較的良好とされる fineweb-edu コーパス [13] を選択した．その際のデータ量は，Chinchilla ルール [14] に従って，前述のモデルパラメータ数である 135M のおよそ 20 倍に相当する 2.7B トークンとした．

語彙数　事前学習時の語彙数としては，18k および 35k の 2 つの設定を採用した．こちらのこちらの値は，図 1 の結果から，被覆率 94.1%および 96.7%の設定であることがわかる．このように学習コーパス中には 1 トークンとなった単語単位のトークンが大半を占める設定である．

4.3　評価タスク

ARC [15] と QASC [16] はいずれも科学知識に関する質問応答データセットである．このうち ARC は，中等教育レベルの「ARC-Easy」と，より難易度の高い「ARC-Challenge」という 2 つのサブセットで構成されている．設問形式に関して，ARC は 4 択式であるのに対し，QASC は 8 つの選択肢から回答を選択する形式となっている．いずれも，英語による知識および推論能力を評価するために広く用いられている代表的なベンチマークデータである．

5　実験結果と考察

5.1　実験結果

QA タスクにおける MP モデルと SPM モデルの性能差を 図 2 に示す．本図は，タスクごとで SPM モデルをベースラインとした場合の正解率の差分を表しており， 値が正のときは MP モデルが SPM モデルを上回る性能を示すことを意味する．横軸は SPM モデルの正解率に対応しており， タスクおよび語彙数ごとに， MP モデルによる性能変化を可視化している．

MP モデルは，サブワード分割を用いず， 単語単位でトークンを割り当てる設計であるため， ベースラインである SPM モデルと比較すると， 同一入力に対するトークン効率の観点では不利である． 一方で，単語とトークンの一対一対応により， 語の意

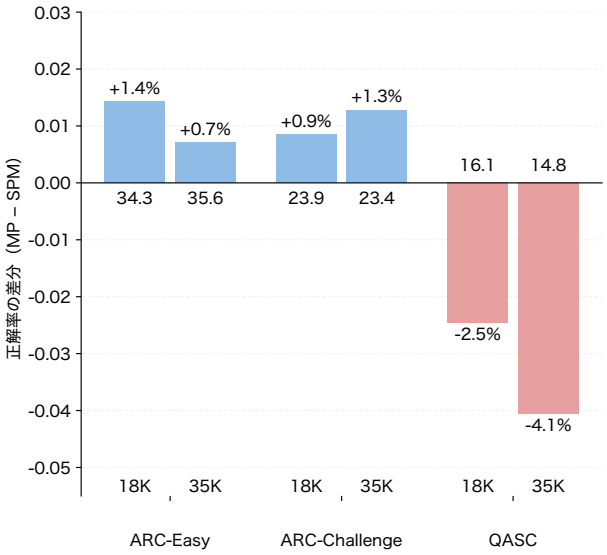

図 2　タスクごとの MP モデルと SPM モデルの性能差.

味単位が分断されずに保持されるという特徴を持つ．その結果，ARC-Easy および ARC-Challenge の 2 タスクにおいては，MP モデルにおいて性能向上が確認された．一方で，すべてのタスクにおいて一貫した性能向上が見られたわけではなく，QASC では MP モデルがベースラインの性能を下回った．

5.2　評価指標

性能差が入力文のトークナイズ挙動の違いとしてどのように現れているかを分析する．本節では MP モデルに着目し，正解サンプルと不正解サンプルを分離した上で，バイトフォールバック率を指標として算出し，その傾向について考察を行う．バイトフォールバック率とは，入力文全体におけるトークン数のうち，バイトレベルトークンが占める割合を指す．バイトフォールバック率が高い場合，本来は語彙中の 1 トークンとして表現されることが期待される単語が，複数のバイト単位のトークン列に分解されて入力されていることを意味する．すなわち，単語という文章の意味構造を判断する際の有力な手がかりが十分に活かせない状態であると解釈できる．

5.3　考察

表 2 に，MP モデルにおけるバイトフォールバック率を正解サンプルと不正解サンプルに分けて示す．いずれのタスクにおいても，不正解サンプルの方が正解サンプルに比べてバイトフォールバック率が高い傾向が確認できる．バイト単位の表現では意味情報が十分に保持されないため、高度な文脈理解を要する QA タスクにおいて、モデルのパフォーマンスが低下する要因となり得る。

表 2　MP におけるバイトフォールバック率.

	MP 18k		MP 35k	
タスク	正解	不正解	正解	不正解
ARC-Easy	0.3105	0.3138	0.1462	0.1528
ARC-Challenge	0.2808	0.2880	0.1666	0.1794
QASC	0.2828	0.2869	0.1920	0.1988

不正解サンプルにおいてバイトフォールバック率が高い傾向が見られたことから，入力中の情報がバイトレベルトークンへと分解されるほど，モデルにとって扱いにくい表現となっている可能性が示唆される．裏返せば，入力が単語（意味単位）として保持され，語彙トークンとして安定的に表現できた場合には，推論に有利に働いたと考えられる．

QASC では，正解サンプルであってもバイトフォールバック率が比較的高く，MP モデルが想定する意味単位での表現が，入力全体に対して必ずしも成立していないことが示唆される．QASC は専門性の高い語彙や複合語が頻出するため，辞書に登録されていない語彙が多く，バイトフォールバックが高い割合で発生したと考えられる．

この結果，単語単位での安定した表現という MP モデルの利点が十分に発揮されにくくなり，性能低下につながった可能性がある．実際に，QASC において MP モデルの性能がベースラインを下回ったという結果は，この傾向と整合する．

6　おわりに

本研究では，サブワード単位のトークン分割が主流の大規模言語モデルに対し，従来一般的であった単語単位のトークン分割を再評価した．その結果，一部のタスクでは推論性能が向上する可能性が示唆された一方，バイトフォールバックが多発する条件では性能低下が確認された．本稿の実験では明確な有効性は確認できなかったため，引き続き多角的に性能を検証していきたい．

付随して，単語単位のトークン分割の利点は，語彙として既知か未知かを明確に判定できる点にある．この性質は，ハルシネーションの抑止などにも寄与する可能性がある．今後はタスク性能に加え，単語単位のトークン分割が有効となる条件についても検討していきたい．

謝辞

本研究は，JST ムーンショット型研究開発事業 JPMJMS2011-35 (fundamental research)，および，文部科学省の補助事業「生成 AI モデルの透明性・信頼性の確保に向けた研究開発拠点形成」の助成を受けたものです．本研究成果の一部は，九州大学情報基盤研究開発センター研究用計算機システムの「一般利用」を利用して得られたものです．

参考文献

[1] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In **Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)**, pp. 1715–1725, 2016.

[2] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao, editors, **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[3] Benjamin Heinzerling and Michael Strube. BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)**, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).

[4] Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. BPE-dropout: Simple and effective subword regularization. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 1882–1892, Online, July 2020. Association for Computational Linguistics.

[5] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[6] William D. Marslen-Wilson and Alan Welsh. Processing interactions and lexical access during word recognition in continuous speech. **Cognitive Psychology**, Vol. 10, No. 1, pp. 29–63, 1978.

[7] Shirley-Ann Rueschemeyer and M. Gareth Gaskell, editors. **The Oxford Handbook of Psycholinguistics**. Oxford University Press, Oxford, 2 edition, 2018.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In **Advances in Neural Information Processing Systems**, Vol. 27, 2014.

[9] Sebastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 1–10. Association for Computational Linguistics, 2015.

[10] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**, pp. 1412–1421. Association for Computational Linguistics, 2015.

[11] Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Smollm-corpus. https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus, 2024. Accessed: 2026-01-08.

[12] Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest collection of educational content, 2024.

[13] Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. **Advances in Neural Information Processing Systems**, Vol. 37, pp. 30811–30849, 2024.

[14] Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. **Advances in Neural Information Processing Systems**, Vol. 36, pp. 50358–50376, 2023.

[15] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. **arXiv preprint arXiv:1803.05457**, 2018.

[16] Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. In **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 34, pp. 8082–8090, 2020.