人間と機械学習のモデルそれぞれに扱いやすいトークン分割に関する実験と考察

平岡 達也　　岩倉 友哉
富士通株式会社
{hiraoka.tatsuya, iwakura.tomoya}@fujitsu.com

概要

本稿では，NLP における機械学習のモデルの性能が向上するようなトークン分割と，人間に読みやすいトークン分割の関係について議論する．JGLUE に含まれる JCommonsenseQA データセットに対して複数のトークン分割を行い，機械学習のモデルによる正解率と，人間のアノテータによる正解率と回答速度を比較する．分析より，人間に読みやすいトークン分割と機械学習モデルの正解率が高くなるトークン分割は必ずしも一致しないことが示唆された．

1　はじめに

トークン分割（Tokenization）は，NLP のシステムの性能や結果に影響を与える重要な前処理である [1, 2, 3]．近年では後段のタスクに応じて，モデルの性能が高くなるようなトークン分割を自動で選択する手法が提案されている [4, 5, 6]．このようなトークン分割は，性能向上という観点から機械学習のモデルに扱いやすいものであると言える．一方で，このトークン分割は必ずしも人間にとって読みやすいものであるとは限らない [5, 6]．人間に読みやすいトークン分割から逸脱したものは，メンテナンスやチューニングの難しさに繋がりうる．例えば，区切りの不一致のために辞書情報が使えなかったり，目視でのエラー分析における解釈の難しさに繋がったりすると考えられる．また，自然言語処理に詳しくないユーザーにとっては，不自然なトークン分割そのものが信頼性低下の原因になりうる．

本研究では，人間と機械のそれぞれに扱いやすいトークン分割について調査を行う．本稿では研究の第一段階として，日本語 QA タスクにおいて異なるトークン分割がもたらす，機械学習のモデルの正解率への影響と，人間にとっての読みやすさへの影響の関係を調査・分析した内容を報告する．

手法	分割例
MeCab	もの_を_入れる_容器_を_なんと_言う_?
Unigram	もの_を入れる_容器_を_なん_と言う_?
BPE	ものを_入れる_容器_を_なん_と言う_?
MaxMatch	ものを_入れる_容器_をな_んと_言う_?
OpTok	もの_を_入れる_容器_を_なん_と言う_?
Random	も_のを入_れ_る_容_器_を_なんと_言う_?

表 1　各トークナイゼーション手法による分割例

2　実験設定

2.1　データセット

本研究では，JGLUE [7] に含まれる JCommonsenseQA データセットを用いて実験を行う．本研究実施時点では JGLUE のテストデータが公開されていないため，学習データと検証データのみを使用する．また，QA モデルに用いる文字分散表現の学習と，トークン分割器の学習には日本語 Wikipedia データを用いた[1)]．

2.2　比較するトークン分割の手法

本研究では，トークン分割の手法として辞書を用いた手法（**MeCab**），教師無しで語彙を構築する手法（**Unigram**，**BPE**，**MaxMatch**），タスクに応じてトークン分割を最適化する手法（**OpTok**），ランダムなトークン分割（**Random**）の 6 つを比較する．

教師なしの手法におけるトークン分割器は，それぞれ語彙の大きさを 64,000 とし，日本語 Wikipedia のタイトルと概要データを用いて学習した．各手法でトークン分割を行ったテキストの例を表 1 に示した．また，実際に JCommonsenseQA の学習データと検証データに対してトークン分割の処理を行い，出現したトークンの種類数を表 2 にまとめた．それぞれの手法について，以下に説明する．

1）https://dumps.wikimedia.org/jawiki/20220820/

	全語彙	QA で使用した語彙
MeCab	-	9,267
Unigram	64,000	11,921
BPE	64,000	12,292
MaxMatch	64,000	12,890
OpTok	64,000	11,440
Random	-	19,077

表 2　トークナイゼーション手法ごとの語彙の規模

MeCab:　辞書を用いたトークン分割（単語分割）手法として，MeCab [8] を用いた．単語辞書には，UniDic [9] を用いた．人手で整備された辞書を用いている本手法は，比較する手法間で最も人間に読みやすいトークン分割であると考えられる．

Unigram:　ユニグラム言語モデルを用いたトークン分割の手法として，SentencePiece [10] のユニグラムモードを使用した．

BPE:　Byte Pair Encoding を用いたトークン分割手法 [11] には，SentencePiece の BPE モードを用いた．

MaxMatch:　最長一致法を用いたトークン分割手法 [12] として，BertTokenizer (WordPiece) を用いた．

OpTok:　データセットとモデルに対して正解率が向上するようなトークン分割を自動的に獲得する手法として，OpTok [6] を利用した．ユニグラム言語モデルでのトークン分割を用いて学習した QA モデル（後述する BiLSTM を用いたもの）に対して，後処理としてトークン分割を最適化した．そのため，語彙はユニグラム言語モデルと一致する．

Random:　単語の長さをサンプリングし，テキストを前から順にランダムに分割した．単語の長さは Wikipedia を MeCab で分割した際の分布に従った．

2.3　QA モデル

機械学習のモデルを用いた実験には，Bag-of-Words [13] と BiLSTM [14, 15] の 2 種類の単純な QA の手法を用いる．N 個のトークンからなる問題文 $q = w_1, ..., w_N$ について，選択肢の単語 a が解答となる確率 $p(a|q)$ を次のように計算する（図 1）．

$$\mathbf{v}_q = g(f(\mathbf{v}_{w_1}, ..., \mathbf{v}_{w_N})), \tag{1}$$

$$p(a|q) \sim \mathbf{v}_q^\top \mathbf{v}_a. \tag{2}$$

ただし，$f(\cdot)$ はエンコーダー，$g(\cdot)$ はエンコーダーの出力をトークンベクトルの次元数に変換する MLP である．また，$\mathbf{v}_{w_n}$ と $\mathbf{v}_a$ はいずれもトークンベクトルである．エンコーダー $f(\cdot)$ の処理は Bag-of-Words を用いる場合（f_{BoW}）と，BiLSTM（f_{BiLSTM}）を用いる場合でそれぞれ以下のように異なる．

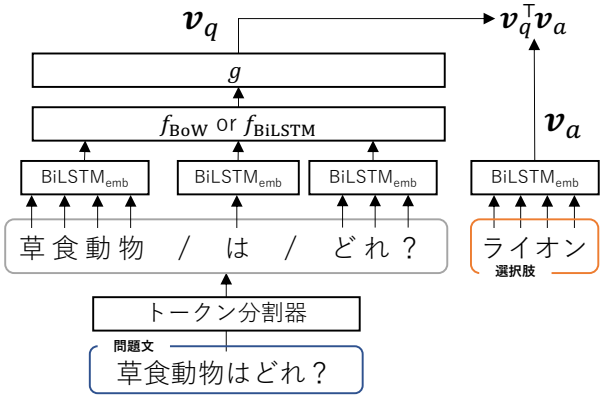

図 1　QA モデルの概要

$$f_{\text{BoW}}(\mathbf{v}_{w_1}, ..., \mathbf{v}_{w_N}) = \frac{\sum_{w \in s} \mathbf{v}_w}{N}, \tag{3}$$

$$f_{\text{BiLSTM}}(\mathbf{v}_{w_1}, ..., \mathbf{v}_{w_N}) = h(\text{BiLSTM}_{\text{QA}}(\mathbf{v}_{w_1}, ..., \mathbf{v}_{w_N})), \tag{4}$$

ここで，$h(\cdot)$ は各出力への最大プーリングである．

トークン分割の差以外の要素による影響を減らすため，トークンベクトルは文字ベクトルから計算する [16, 17]．M 文字のトークン $w = c_1, ..., c_M$ のベクトル $\mathbf{v}_w$ は，次のように計算する．

$$\mathbf{v}_w = h(\text{BiLSTM}_{\text{emb}}(\mathbf{v}_{c_1}, ..., \mathbf{v}_{c_M})). \tag{5}$$

ここで $\mathbf{v}_{c_m}$ は c_m の文字分散表現である．$\mathbf{v}_{c_m}$ と $\text{BiLSTM}_{\text{emb}}$ は，日本語 Wikipedia の全文を言語モデルとして学習し [18]，QA の学習時には固定した．g と f_{BoW}，f_{BiLSTM} は，QA の学習データで学習した．

ハイパーパラメータについて，文字分散表現 $\mathbf{v}_c$ のサイズは 256，$\text{BiLSTM}_{\text{emb}}$ と $\text{BiLSTM}_{\text{QA}}$ の隠れベクトル，$\mathbf{v}_w$ のサイズは 1,024，各 BiLSTM のレイヤ数は 1 とした．交差エントロピー誤差を用いて Adam [19] で更新を行い，30 エポックの学習のうち検証データでの性能が最も高いモデルを採用した．

2.4　人間による読みやすさの収集

2.4.1　読みやすさの順位付け

異なるトークン分割に対する読みやすさの最も単純なアノテーション方法として，人間による順位付けを行った．実際のアノテーション画面を図 2 に示した．アノテータは，ランダムに並べられた最大 6 つのトークン分割について，区切り方が適切だと思うものから順に並び変えるよう指示される．本作業は 1 名のアノテータが担当した．

図 2　読みやすさの順位付けアノテーション画面

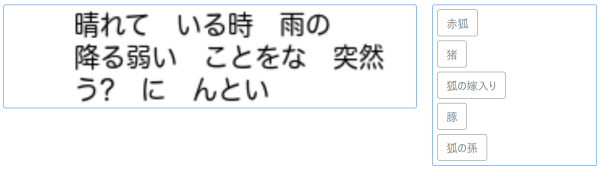

図 3　Bag-of-Words によるアノテーション画面

2.4.2　Bag-of-Words での回答収集

2.4.1 節のアノテーション方法では，適切な区切り方の基準がアノテータの知識に強く依存する．本研究では，タスクを解くうえで各トークン分割がどの程度読みやすいかの情報を収集したい．そのため，この方法では目的が達成できるとは言えない．

そこで，異なるトークン分割を用いたときの QA タスクへの回答時間及び正解率を収集することで，人間に読みやすいトークン分割を分析する[2]．具体的には，人間に読みやすいトークン分割は，語順がランダムであっても元の意味を復元しやすいと仮定し，図 3 のようなアノテーション画面を用いて，Bag-of-worrds 形式での回答を行った．

アノテーション時には，各設問画面の表示から最後に選択肢のボタンを押下するまでの時間を記録する．トークン分割が読みやすいものである場合，この時間が短くなる傾向が見られると期待される．なお，クラウドワーカーによるアノテーションを行っているため，作業中の行動のすべてを監視することは出来ない．そのため，回答時間が 30 秒を超えるもの[3]については，画面を開いた状態で放置した可能性が高いとものとし，集計の対象から除外した．

本作業では，1 名のアノテータが問題ごとにすべてのトークン分割の種類について回答する．そのため，異なるトークン分割で問題文を提示しているとはいえ，過去に解いたことのある問題に対して何度か回答を行うことになる．そこで，問題の丸暗記を防ぐために，同じ問題の間隔をある程度あけて（平均 1591.8 件）提示するようにした．また，回答内容の記憶を防ぐために，アノテータへの正答の提示も行わない．本作業には 6 名のアノテータが参加しており，現時点で全サンプル 10,058 件について各 1 名の回答結果が収集できている。

2）　画像とテキストを用いた既存調査 [20] に倣った．
3）　図 3 の例から分かる通り，平均的な回答時間は 5 秒程度，設問が長いものでも 15 秒程度で回答可能である．

	Bag-of-Words	BiLSTM
MeCab	42.48	44.06
Unigram	43.97	**44.80**
BPE	**44.15**	44.53
MaxMatch	43.61	44.74
OpTok	43.37	44.30
Random	41.94	42.39

表 3　JCommonsenseQA での正解率（検証データ）．列内の**最大値を太字**，最小値を下線でハイライトした．

3　結果と分析

3.1　読みやすさと機械学習モデルの正解率

QA モデルを用いたときの，各トークン分割手法ごとの正解率を表 3 に示した[4]．Bag-of-Words を用いたモデルでは BPE，Unigram の順に正解率が高く，BiLSTM を用いたモデルでは Unigram，MaxMatch の順に正解率が高い．また，どちらもモデルも Random，MeCab の順に正解率が低くなっている．

表 4 の「読みやすさ」の列には，2.4.1 節の方法で収集した読みやすさの順位付けについて，各トークン分割手法が 1 位となった割合を示した．なお，予稿投稿時点では全サンプルのうち，学習データの 3,302 件のみアノテーションが完了している．結果より，人手で整備した辞書を用いている MeCab が最も読みやすいことが分かる．一方で，MeCab によるトークン分割は，機械においては Random の次に正解率が低いものである．ここから，人間に読みやすいトークン分割は，必ずしも機械の正解率が高いものとは一致しないと言える．

4）　3 回試行の平均値を報告．OpTok については，ある学習済みのモデルに対して最適化したトークン分割を，初期化が異なる別の QA モデルの学習に再利用し，3 回試行を行った．そのため，OpTok を用いた場合の正解率は必ずしも Unigram よりも高い値にならない．

	読みやすさ	正解率			平均回答時間			平均トークン数		
	学習	学習	検証	全体	学習	検証	全体	学習	検証	全体
MeCab	**68.23**	**93.90**	**97.50**	**94.30**	5.44	5.09	5.40	9.7	7.04	9.61
Unigram	16.38	93.85	97.4	94.25	5.37	5.02	5.33	7.66	5.78	7.59
BPE	5.97	**93.90**	97.32	94.28	5.44	5.06	5.40	7.4	5.54	7.34
MaxMatch	4.12	93.69	**97.50**	94.11	5.42	5.17	5.39	7.35	5.55	7.29
OpTok	4.09	93.69	97.05	94.06	5.35	5.04	5.31	7.98	5.88	7.91
Random	1.21	93.23	97.23	93.67	**6.06**	**5.98**	**6.05**	**10.03**	**7.4**	**9.94**

表 4　各トークン分割手法で問題文を区切って提示した際の，読みやすさ 1 位の割合（%，学習データの 3,302 件）と正解率（%），平均回答時間（単位は秒，正答・回答時間が 30 秒以内のサンプルのみを集計），各サンプルの問題文の平均トークン数．列内の**最大値を太字**，最小値を下線でハイライトした．

3.2　機械学習モデルと人間の正解率

表 4 の「正解率」の列には，2.4.2 節で説明した方法で収集した人間による正解率をまとめた．データ全体での人間の正解率は MeCab，BPE の順に高い．また，正解率は Random，OpTok の順に低い．

これらの内容から，ランダムなトークン分割は人間にも機械にも扱いにくいと言える．また，人間と機械でトークン分割による正解率の大小関係が異なることから，人間と機械の正解率が高いトークン分割が必ずしも一致しないと言える．

3.3　分割と回答時間の関係

表 4 の「平均回答時間」からは，OpTok，Unigram の順に回答時間が短いことが分かる．OpTok は Unigram をベースとして最適化を行ったトークン分割であるため，人間は言語モデルを用いて語彙の構築とトークン分割を行ったテキストを効率的に読むことができると言える．OpTok は Unigram によるトークン分割のうち，助詞などの高頻度語を更に切り離す傾向がある [6]（表 1）ため，今回の提示形式において語順の推測がしやすくなり回答時間が短くなっていると考えられる．一方で Random なトークン分割を用いたときの回答時間は顕著に長く，読みづらいトークン分割と言うことができる．

MeCab を用いたときの回答時間は，Random の次に長い．これは，MeCab によるトークン分割が Unigram などに比べて細かく，問題文を表すためのトークン数が多いことに起因すると考えられる（表 4「平均トークン数」）．トークン数が多い場合，内容を把握するために時間がかかることから，回答時間も長くなると考えられる．実際に全アノテーション内容についての，問題文に含まれるトークン数と回答時間との相関係数は 0.22 であり，弱い正の相関がある．一方で，平均トークン数が比較的多い OpTok の回答時間が最も短いことから，トークン数以外の要素が回答時間の長さに影響を与えていると考えられる．例えば，MeCab を用いる場合は単語の意味が理解できることから，より内容について長考する傾向にある，などの原因があると考えられる．

4　おわりに

本稿では，人間と機械学習のモデルのそれぞれに扱いやすいトークン分割について，調査と実験の結果を報告した．QA のデータを用いた実験では異なるトークン分割手法を用いて問題文を分割し，機械による正解率を収集した．また，問題文の語順をシャッフルした上で人手でのアノテーションを行い，人間による回答の精度と回答時間を収集した．収集した結果より，機械学習のモデルの正解率が高くなるようなトークン分割は，必ずしも人間による正解率が高くなるものや，人間に読みやすいものに一致するとは限らないことが示唆された．

本研究では引き続きアノテーション作業を進め，人間による正解率や回答時間に関する情報を収集する．また，トークン分割の差が人間の回答に及ぼす影響の調査方法については，現在の方法では差が大きく表れていないことから，方法の改善が必要である．例えば，問題文に含まれるトークンをフラッシュカード形式で提示した場合の正解率や，アノテータの視線情報を用いたデータ収集などが考えられる．機械学習のモデル側についても，実験設定の改善の余地は大いにある．例えば，より現代的な比較を行うためには，大規模言語モデル [21] の事前学習から異なるトークン分割手法を用い，その影響を調査する必要がある．今後も実験内容の再検討を重ね，人間と機械学習のモデルに扱いやすいトークン分割の差について調査を行う．

謝辞

本研究は，JST，ACT-X，JPMJAX21AM の支援を受けたものです．また，アノテーションの設計では熊野康孝氏と六条範俊氏にご助言いただきました．

参考文献

[1] Tatsuya Hiraoka, Hiroyuki Shindo, and Yuji Matsumoto. Stochastic tokenization with a language model for neural text classification. In **Proceedings of the 57th Annual Meeting of ACL**, pp. 1620–1629, 2019.

[2] Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. In **Findings of ACL: EMNLP 2020**, pp. 4617–4624, 2020.

[3] Hoo-Chang Shin, Yang Zhang, Evelina Bakhturina, Raul Puri, Mostofa Patwary, Mohammad Shoeybi, and Raghav Mani. BioMegatron: Larger biomedical domain language model. In **Proceedings of the 2020 Conference on EMNLP**, pp. 4700–4706, Online, November 2020. Association for Computational Linguistics.

[4] Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. Dynamic programming encoding for subword segmentation in neural machine translation. In **Proceedings of the 58th Annual Meeting of ACL**, pp. 3042–3051, Online, July 2020. Association for Computational Linguistics.

[5] Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. Optimizing word segmentation for downstream task. In **Findings of ACL: EMNLP 2020**, pp. 1341–1351, Online, November 2020. Association for Computational Linguistics.

[6] Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. Joint optimization of tokenization and downstream model. In **Findings of ACL: ACL-IJCNLP 2021**, pp. 244–255, 2021.

[7] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In **Proceedings of the Thirteenth LREC**, pp. 2957–2966, Marseille, France, June 2022. European Language Resources Association.

[8] Taku Kudo. Mecab: Yet another part-of-speech and morphological analyzer. http://taku910.github.io/mecab/, 2006.

[9] 康晴伝, 智信小木曽, 秀樹小椋, 篤山田, 信明峯松, 清貴内元, 花絵小磯. コーパス日本語学のための言語資源：形態素解析用電子化辞書の開発とその応用. 日本語科学, Vol. 22, pp. 101–123, 10 2007.

[10] 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 EMNLP: System Demonstrations**, pp. 66–71, 2018.

[11] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In **Proceedings of the 54th Annual Meeting of ACL**, Vol. 1, pp. P1715–1725, 2016.

[12] Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast wordpiece tokenization. In **Proceedings of the 2021 Conference on EMNLP**, pp. 2089–2103, 2021.

[13] Jordan Boyd-Graber, Brianna Satinoff, He He, and Hal Daumé III. Besting the quiz master: Crowdsourcing incremental classification games. In **Proceedings of the 2012 Joint Conference on EMNLP and CoNLL**, pp. 1290–1301, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. **Neural computation**, Vol. 9, No. 8, pp. 1735–1780, 1997.

[15] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In **Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.**, Vol. 4, pp. 2047–2052. IEEE, 2005.

[16] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In **Proceedings of the 2016 Conference of NAACL: HLT**, pp. 260–270, 2016.

[17] Chunqi Wang and Bo Xu. Convolutional neural network with word embeddings for chinese word segmentation. In **Proceedings of the Eighth IJCNLP**, pp. 163–172, 2017.

[18] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In **Proceedings of the 2018 Conference of NAACL: HLT, Volume 1 (Long Papers)**, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

[20] Chunpeng Ma, Aili Shen, Hiyori Yoshikawa, Tomoya Iwakura, Daniel Beck, and Timothy Baldwin. On the effectiveness of images in multi-modal text classification: An annotation study. **ACM Trans. Asian Low-Resour. Lang. Inf. Process.**, oct 2022. Just Accepted.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.