# 転移学習における強化学習を用いた効率的なトークナイザとモデルの同時学習

平子潤 [1]　柴田知秀 [2]
[1] 名古屋大学大学院　[2] ヤフー株式会社
hirako.jun@g.mbox.nagoya-u.ac.jp　tomshiba@yahoo-corp.jp

## 概要

文をトークンの系列に分割するトークナイズは、言語処理の最初で行われる重要なステップである。本研究では、転移学習において、より計算コストの増加が小さい、効率的なトークナイザとモデルの同時学習手法を提案する。提案手法は、トークナイズ確率を方策、負のモデルロスを報酬とした強化学習によりトークナイザを学習する。実験の結果、提案手法は既存手法よりも高い性能を達成し、また、計算時間を削減できたことを確認した。

## 1　はじめに

文をトークンの系列に分割するトークナイズは、言語処理の最初で行われる重要なステップである。特に日本語のような単語が空白で区切られていない言語では、適切な単位にトークナイズすることが重要となっている。一般に、トークナイザはモデルとは独立に事前に定義して、モデルの学習・推論時は固定することが一般的であり、タスクに最適なトークナイザである保証はない。

この問題に対して、平岡らはモデルの学習と同時にトークナイザの学習も行う手法を提案している [1]。この手法の基本的な考えは、いくつかのトークナイズ結果を出力し、タスクがうまく解けるようなトークナイズ結果の確率が高くなるように、トークナイザを学習することである。モデルを学習するための入力に加えて、トークナイザを学習するために $N$ 回[1)]モデルにトークナイズ結果を通す必要があり、比較的計算コストが大きい手法となっている。この手法は転移学習以前の手法において有効性が検証されており、BERT [2] などの転移学習モデルに適用する、というのは自然な考えであるが、事前学習時からトークナイザを同時学習することは計算コストの観点から難しい。

この問題を解決するため、本研究では、転移学習にも適用できるように、より計算コストの増加が小さい、効率的なトークナイザとモデルの同時学習手法を提案する。提案手法は、強化学習における方策勾配法を用い、トークナイズ確率を方策、負のモデルロスを報酬とし、報酬を最大化、つまりモデルのロスを最小化するようにトークナイザの学習を行う。具体的には、1-best tokenize した結果と、sampling tokenize した結果を比較し、前者をモデルに入力した時のモデルのロスに比べ、後者をモデルに入力した時のモデルのロスが小さければ、後者のトークナイズ確率が大きくなるようにトークナイザを学習する。先に述べた平岡ら手法と提案手法の概要を図 1 に示す。どちらの手法もモデルの学習には sampling tokenize の結果を用いる点では同一である (それぞれの手法の図における右側) が、トークナイザの学習に用いるトークナイズが、平岡ら手法では N-best であるのに対し、提案手法では 1-best 一つのみであるため、平岡ら手法に比べて計算量を大きく削減することができる。

日本語言語理解ベンチマークで評価を行い、提案手法が比較手法よりも精度が高いことを示し、また、平岡ら手法を事前学習にも適用した手法よりも計算時間を 50%以下に改善できることを確認した。

## 2　提案手法

提案手法では、事前学習を行ってからファインチューニングを行う転移学習において、モデルとトークナイザを同時に学習する。計算コストが大きい事前学習に対して適用するために、強化学習を用いた定式化を行う。

1)　実験では N=3 などの値が用いられている。

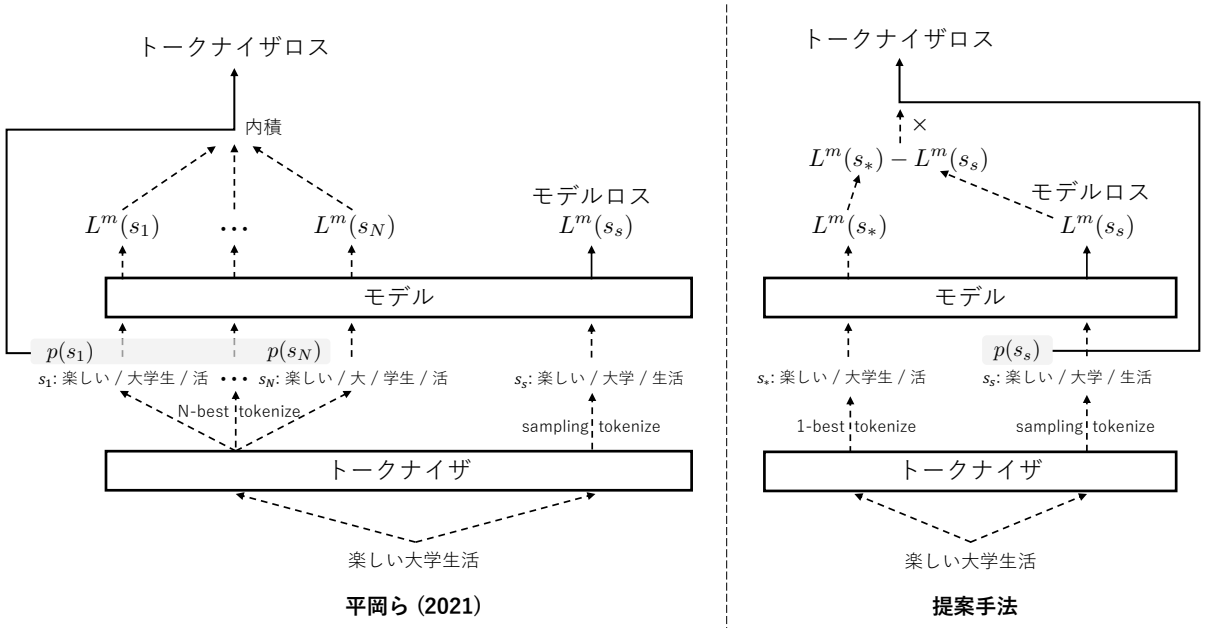


**図 1**　提案手法 (右) と平岡ら手法 (左) の概要 (実線は計算グラフを構築し、誤差逆伝播を行うパスであることを示す)

## 2.1　トークナイザ

学習可能なトークナイザとして、平岡らと同様にニューラルユニグラム言語モデルを用いたトークナイザを利用する。このトークナイザでは、まず、語彙中の各トークンにスコア $score_t$ を割り当て (この値を学習する)、単語横断的に softmax をとることでユニグラム確率 $p(t)$ を計算する。

文 $S$ が $s = (t_1, \ldots, t_k, \ldots, t_K)$ ($K$ はトークン数) に分割されるとき、このように分割される確率 $p(s)$ はトークン列 $s$ に含まれるトークン $t_k$ のユニグラム確率の積で計算する。

本研究では、このユニグラム言語モデルを用いたトークナイザで、1-best tokenize と sampling tokenize の 2 つのトークナイズ結果を使用し、これらの 2 つのトークナイズ結果それぞれをモデルに入力した時のロスを比較しながらトークナイザを学習する。1-best tokenize と sampling tokenize を行うための、n-best tokenize は文 $S$ に含まれる、あらゆるユニグラム確率 $p(t)$ に対して Forward-DP Backward-A*アルゴリズム [3] を適用することで計算する。

## 2.2　モデルとトークナイザの同時学習

モデルの学習では、直接的な教師が存在する一方で、トークナイザには直接的な教師が存在せず、適切に教師を設定する必要がある。そこで、タスクにおけるモデルのロスをトークナイザの間接的な教師とし、モデルのロスが小さくなるような単語分割をおこなうようにトークナイザを学習する。トークナイザからモデルへのパスは微分不可能であるため、強化学習を用いてトークナイザを学習する。

### 2.2.1　強化学習の枠組み

強化学習の枠組みとして、文生成タスクに使われている強化学習アルゴリズム Self-critical Sequence Training for Image Captioning (SCST) [4] をベースとする。SCST は REINFORCE アルゴリズム [5] をベースとし、単語生成確率を方策、生成された文に対するスコアを報酬とする。greedy に生成した文の報酬と sampling して生成した文の報酬を比べ、後者の方が高ければ sampling した単語の生成確率を高めるように学習される。

提案手法では SCST における greedy に生成した文の代わりに **1-best tokenize した文** $s_*$、sampling で生成した文の代わりに **sampling tokenize した文** $s_s$ を利用し、報酬としては負のモデルのロスを利用する[2)]。$s_*$ ならびに $s_s$ をモデルに入力した時のロスの大きさを比較し、$s_*$ のモデルロスに比べて $s_s$ のモデルロスが小さければ、sampling tokenize で得られたトークナイズ結果の方がモデルがタスクを解きやすいということになるので、サンプリング系列のトークナイズ確率を上げるように、トークナイザを学習する。ここでモデルとは事前学習時はマスク言

2)　提案手法では SCST のように時系列にそって行動 (単語の生成) を選択するわけではなく、一度行動 (トークナイズ) を選択するのみである。

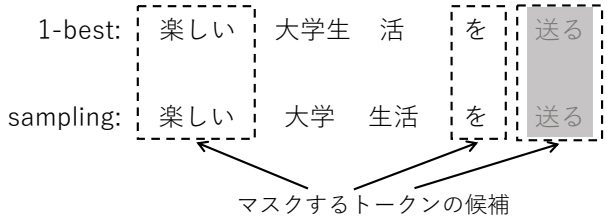


**図 2** マスク言語モデルによって事前学習を行う際のマスクするトークンの候補の選択方法

語モデル、ファインチューニング時は下流タスクのモデル (文分類タスクの場合のモデルなど) を指す。

トークナイザのロス関数 $L^t$ は以下のように与えられる。

$$L^t = -(r(s_s) - r(s_*))log\ p_\theta(s_s) \quad (1)$$

ここで、$\theta$ はトークナイザのパラメータ、報酬 $r$ は負のモデルロス $-L^m$ を表わす。モデルのパラメータの更新は、$s_s$ をモデルに入力した時のロスを用いて行い、このモデルロスをトークナイザの更新 (式 (1) の $r(s_s)$ が負のモデルロスに相当) にも利用する。$s_s$ でモデルのパラメータを更新することによって、サブワード正則化 [6] を行うことができ、性能向上が期待される。

#### 2.2.2 事前学習とファインチューニング

**事前学習** マスク言語モデルによって事前学習を行う場合、通常、入力トークン列からマスクするトークンをランダムに選択するが、提案手法の場合、1-best tokenize と sampling tokenize ではトークナイズ結果が異なり得ることから、注意が必要となる。これらの 2 つのトークナイズ結果において、マスク言語モデルのロスを平等に比較できるように、マスクするトークンの候補は、1-best tokenize と sampling tokenize で同様に分割されたトークンに限定する。図 2 の例ではマスクするトークンの候補は「楽しい」「を」「送る」となり、この例では「送る」がマスクするトークンとして選ばれている。

**ファインチューニング** ファインチューニングでも同様に、1-best tokenize と sampling tokenize でのモデルロスを比較し、sampling tokenize を用いた場合のモデルロスが小さければ、そのサンプリングのトークナイズ確率を上げるようにトークナイズを学習する。推論時は 1-best tokenize の結果を用いる。

### 2.3 既存手法との計算コストの比較

図 1 に示すように、提案手法と平岡ら手法では、トークナイザを学習するためにモデルに入力するトークナイズ結果の数が異なっており、計算コストに差が生まれている。まず、図の左に示す平岡ら手法では、トークナイザを学習するために、モデルの学習に用いるトークナイズ結果とは別に N 個のトークナイズ結果をモデルに入力する必要がある[3)]。

一方で、図の右に示す提案手法ではトークナイザを学習するために追加でモデルに入力するのは、$s_*$ のみである。さらに、トークナイズの段階で $s_s$ と $s_*$ が一致した場合は、トークナイザの学習を行なわず、$s_*$ をモデルに入力する必要がないため、さらに計算コストを減らすことができる。以上より、提案手法を用いることで、既存の平岡ら手法よりも、小さい計算コストでモデルとトークナイザの同時学習を行うことが可能となる。

## 3 実験

複数の分類タスクを用いて提案手法の有効性を検証した。

### 3.1 設定

#### 3.1.1 データセット

日本語言語理解ベンチマークである JGLUE [7] を用いて評価を行った。JGLUE のうち、文章分類、文ペア分類のタスクである MARC-ja、JSTS、JNLI を利用した[4)]。各種データセットの詳細は付録 B に示す。

#### 3.1.2 比較手法

以下の 4 つのモデルと提案手法を比較した。

**UniLM** ユニグラム言語モデルを用いたトークナイザを利用して、BERT の事前学習とファインチューニングを行う手法。ソフトウェアとしては SentencePiece [8] を用いた。このモデルは、事前学習、ファインチューニングともにトークナイザは学習されない。

**平岡ら手法** 1 節で述べたとおり、平岡ら手法は転移学習以前のモデルで評価されているが、提案手法との比較のために、BERT のファインチューニングにおいて、N-best トークナイズを用いてモデルとトークナイザを同時に最適化する手法をここでは平

3) N-best の入力間でロスの大きさを比較するため、N は 2 以上である必要がある。

4) JGLUE に含まれるタスクのうち、JSQuAD は形態素区切と Sentencepiece での区切が一致しないため、また、JCommonsenseQA は質問ならびに選択肢の文長が非常に短くトークナイズの影響が小さいため、これらのタスクは提案手法での評価からは除いた。

表 1　JGLUE の文章分類・文ペア分類タスクを用いた評価実験の結果

| | モデル | トークナイズ学習 事前学習 | トークナイズ学習 ファインチューニング | MARC-ja acc | JSTS Pearson / Spearman | JNLI acc | 平均 |
|---|---|---|---|---|---|---|---|
| (1) | UniLM | - (22.8%) | - | 95.71 ±0.19 | 88.80 ±0.09 / 84.77 ±0.16 | 88.67 ±0.08 | 90.39 |
| (2) | 平岡ら手法 | - (22.8%) | ✓ | **95.82** ±0.13 | 88.09 ±0.30 / 83.85 ±0.22 | 88.10 ±0.52 | 89.96 |
| (3) | 提案手法 (N-best tokenize) | ✓ (100%) | ✓ | 94.85 ±0.31 | 86.80 ±0.16 / 82.94 ±0.14 | 87.17 ±0.45 | 88.96 |
| (4) | 提案手法 (強化学習; FT のみ) | - (22.8%) | ✓ | 95.79 ±0.14 | 88.40 ±0.25 / 84.28 ±0.30 | 88.48 ±0.28 | 90.20 |
| (5) | 提案手法 | ✓ (48.8%) | ✓ | 95.79 ±0.06 | **89.28** ±0.38 / 84.80 ±0.47 | **89.09** ±0.19 | **90.64** |

岡ら手法と呼ぶ。この手法は再現実装した。

**提案手法 (N-best tokenize)**　平岡ら手法で採用されている N-best トークナイズを用いて、事前学習時にもトークナイザを学習する手法[5)]。

**提案手法 (強化学習; FT のみ)**　BERT の事前学習時はトークナイザを固定し、ファインチューニング時のみ、強化学習を利用する提案手法でモデルとトークナイザを同時に学習する手法。

**提案手法**　BERT の事前学習時とファインチューニング時のどちらも、強化学習でモデルとトークナイザを同時に学習する手法。

### 3.1.3　モデル学習の設定

提案手法を含む全ての比較モデルで、日本語 Wikipedia を用いて、BERT の事前学習を行った[6)]。トークナイザのハイパーパラメータについては付録 C に示す。ファインチューニングでは、損失関数として、分類タスクである MARC-ja と JNLI では交差エントロピー損失、回帰タスクである JSTS では平均二乗誤差を利用した。ファインチューニングはシードを変えて 3 回実行し、各評価尺度の平均と標準偏差を報告する。また全体的な性能を比較するために、3 つのタスクの性能の平均も算出した[7)]。

## 3.2　実験結果と考察

実験結果を表 1 に示す。全体的な性能は比較手法と比べ、提案手法が最も高いことを確認した。(2) もしくは (4) と (5) を比べることによって、事前学習もトークナイズ学習を行った方がよいことがわかり、(3) と (5) を比べることによって、平岡らが採用している N-best tokenize よりも提案手法の強化学習を用いた方がよいことがわかる。

次に、提案手法を用いることで、平岡ら手法を用いて事前学習を行った場合に比べて、どれくらい事前学習の時間を軽減することができるかについて述べる。提案手法 (N-best tokenize) にかかる時間を 100%とすると、提案手法が 48.8%、UniLM が 22.8%となった。この結果から、提案手法 (N-best tokenize)、すなわち、平岡ら手法をそのまま事前学習に適用した場合と比べ、提案手法は 2 倍以上計算時間が改善できることを確認できた。

表 2　MARC-ja における改善例 (正解ラベルは P)

| モデル | 出力ラベル | トークナイズ結果 |
|---|---|---|
| UniLM | N | 今 はこれ がない とも の たり ない です ね 。 |
| 平岡ら手法 | P | 今 はこれ **がないとも** たり ない **です** ね 。 |
| 提案手法 | P | 今 **はこれが ないとも** たり ない です ね 。 |

MARC-ja における改善例を表 2 に示す。MARC-ja の例に含まれる「これがないとものたりない」は「これがあると満足ができる」というポジティブなフレーズであるが、そのことを理解するためには、「これが」、「ないと」「ものたりない」をそれぞれ理解する必要がある。しかしながら、UniLM では、「これ」、「がない」、「とも」、「の」、「たり」、「ない」のように、意味を汲むためのトークンの境界と一致しておらず、適切に意味を捉えることができていない。一方で、提案手法では、「これが」、「ないと」、「もの」、「たり」、「ない」のように、トークンの境界が一致しており、モデルが適切に意味を捉えることができるようになっていると考えられる。

# 4　おわりに

本論文では転移学習において、トークナイザとモデルの同時学習する手法を提案した。強化学習を用いて定式化することにより、計算量を抑えられ、かつ、精度が高いことを示した。今後の課題としては、本研究で採用したトークナイザは各サブワードの生成確率のみを学習するものであり、提案手法が比較手法に比べて大幅な精度向上を達成できなかったのは、このトークナイザの表現力がそれほど高くないことが考えられ、より表現力の高いトークナイザを検討する予定である。

5)　事前学習でトークナイズ学習を行っているという意味で、提案手法の亜種とみなしている。
6)　ただし、next sentence prediction は文献 [9] で有効でないとされていることから行っていない。
7)　JSTS の Peason と Sperman の値を平均してから、MARC-ja、JSTS、JNLI の性能の平均をとっている。

# 参考文献


[1] Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. Joint optimization of tokenization and downstream model. In **Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021**, pp. 244–255, Online, August 2021. Association for Computational Linguistics.

[2] 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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] Masaaki Nagata. A stochastic Japanese morphological analyzer using a forward-DP backward-A* n-best search algorithm. In **COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics**, Kyoto, Japan, August 1994.

[4] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, July 2017.

[5] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. **Machine Learning**, Vol. 8, pp. 229–256, 1992.

[6] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In **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.

[7] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In **Proceedings of the 13th Language Resources and Evaluation Conference**, Marseille, France, 2022. European Language Resources Association.

[8] 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, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[9] 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, 2019.

[10] Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. The multilingual Amazon reviews corpus. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 4563–4568, Online, November 2020. Association for Computational Linguistics.

[11] Takashi Miyazaki and Nobuyuki Shimizu. Cross-lingual image caption generation. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1780–1790, Berlin, Germany, August 2016. Association for Computational Linguistics.

**表 3** JSTS・JNLI における改善例

| モデル | 出力 | トークナイズ結果 |
|---|---|---|
| JSTS (正解スコア: 3.4) | | |
| UniLM | 0.7 | 子供が スパ ゲ ッ ティー を 高く 掲げ ています 。 [SEP] 小さな 女の子 が 皿 の中の 麺 を持ち 上げ ています 。 |
| 平岡ら手法 | 1.6 | 子供が スパ ゲ ッ ティー を 高く 掲げ ています 。 [SEP] 小さな 女の子 が 皿 の中の 麺 を持ち 上げ ています 。 |
| 提案手法 | 2.1 | **子供** が スパ ゲ ッ ティー を 高く 掲げ ています 。 [SEP] 小さな 女の子 が 皿 の中の 麺 を持ち 上げ ています 。 |
| JNLI (正解ラベル: C) | | |
| UniLM | E | 踏み 切り の中 を 乳 母 車が 通行 している ところ な んです 。 [SEP] 踏み 切り の中 を 車が 通行 している ところです 。 |
| 平岡ら手法 | C | 踏み 切り **の 中** を 乳 母 車が 通行 している ところ な んです 。 [SEP] 踏み 切り **の 中** を 車が 通行 している ところです 。 |
| 提案手法 | C | 踏み 切り の中 を 乳 母 **車 が** 通行 している ところ **なん です** 。 [SEP] 踏み 切り の中 を **車** が 通行 している **ところ です** 。 |

# A JSTS・JNLI における改善例

JSTS と JNLI における改善例を表 3 に示す。UniLM ではタスクを解くために重要なトークンが助詞とくっついてしまっている。JSTS の例では「子供が」のように「子供」と助詞「が」がくっついており、前提文の「子供」と仮説文の「女の子」の関係を捉えるのが難しい。一方、提案手法ではそれらのトークンが、助詞と離れており、より意味を適切に捉えることができるようになっていると考えられる。

# B JGLUE のデータセットの詳細

MARC-ja は、Multilingual Amazon Reviews Corpus [10] の、日本語レビューで構築したデータセットであり、各レビューをポジティブとネガティブの 2 値で分類するタスクを解く。JSTS は、YJ Captions Dataset [11] を基に構築された、意味的類似度計算データセットであり、文ペア間の意味的な類似度を 0 (意味が完全に異なる) から 5 (意味が等価) で回帰予測するタスクを解く。JNLI は、JSTS で同じテキストで構築された自然言語推論データセットであり、文ペアに対して、前提文が仮説文に対して持つ推論関係を含意、矛盾、中立の 3 値で分類するタスクを解く。評価指標としては、MARC-ja、JNLI では accuracy を用い、JSTS では Pearson および Spearman の相関係数を用いる。また、各データセットのデータ数を表 4 に示す。

**表 4** 各データセットのデータ数

| データセット | train | dev |
|---|---|---|
| MARC-ja | 187,528 | 5,654 |
| JSTS | 12,463 | 1,457 |
| JNLI | 20,117 | 2,434 |

# C トークナイザのハイパーパラメータ

提案手法で用いているトークナイザには、N-best の N と、sampling tokenize の確率分布を制御するための $\alpha$ の 2 つのハイパーパラメータがある。この 2 つのパラメータを変化することで、提案手法の性能がどのように変化するかを調査した。N を {4, 16}、$\alpha$ を {0.2, 1.0} でそれぞれ変化させた。計算コストの観点から、広範囲に探索することはできなかった。結果を表 5 に示す。N=4, $\alpha$=1.0 のときが最も性能が高いという結果となった。このハイパーパラメータは、sampling tokenize するときに、確率がより上位のトークナイズ結果をサンプリングするという設定である。この結果から、提案手法において、確率が上位であるトークナイズ結果同士でロスの比較を行うことがより良いトークナイザの構築に重要であり、確率が下位のトークナイズ結果をサンプリングして比較することの有用性は低いことが示唆される。

**表 5** ハイパーパラメータ探索の結果

| N | $\alpha$ | MARC-ja | JSTS | JNLI |
|---|---|---|---|---|
| | | acc | Pearson / Spearman | acc |
| 4 | 0.2 | 95.68 $_{\pm 0.14}$ | 88.98 $_{\pm 0.10}$ / 84.67 $_{\pm 0.16}$ | 88.21 $_{\pm 0.47}$ |
| 4 | 1.0 | 95.79 $_{\pm 0.06}$ | **89.28** $_{\pm 0.38}$ / **84.80** $_{\pm 0.47}$ | **89.09** $_{\pm 0.19}$ |
| 16 | 0.2 | 95.55 $_{\pm 0.07}$ | 88.64 $_{\pm 0.24}$ / 84.38 $_{\pm 0.30}$ | 87.63 $_{\pm 0.21}$ |
| 16 | 1.0 | **95.84** $_{\pm 0.04}$ | 88.27 $_{\pm 0.17}$ / 83.94 $_{\pm 0.12}$ | 87.85 $_{\pm 0.35}$ |