E-006

トークナイズ手法が日本語 word similarity タスクに及ぼす影響について
Effects of tokenization methods on Japanese word similarity task

山口 聖輝† 原 一夫† 鈴木 郁美‡
Toshiki Yamaguchi Kazuo Hara Ikumi Suzuki

1. 概要

トークナイザによる文の分割が異なると、自然言語処理タスクの精度は異なると考えられる。しかし、近年、トークナイザは、BERT や GPT 等の事前学習を前提とする大規模言語モデルと組み合わせて実装されることが多いが、トークナイザが自然言語処理タスクに与える影響を調べた研究は少ない。本研究は、MeCab[1)] による単語分割、Byte-Pair Encoding (BPE) [2, 3] によるサブワード分割、Unigram Language Model (以下、Unigram) [4] によるサブワード分割という、3 つのトークナイザが出力するトークン列を word2vec [6] に与え、得られる単語ベクトルを用いて word similarity タスクの精度比較を行った。その結果、サブワード分割は単語分割よりも良い精度を示した。さらに、サブワード分割後のトークンの出現回数分布を調べると、単語分割後のトークンの出現回数分布に比べて、低頻度語が少ないことが観察された。

2. 本研究の背景

文を構成するトークンの意味を正しく捉えることは、文の意味を正しく理解するために必要と考えられる。一方、文を構成するトークン間に空白のない日本語の自然言語処理においては、文をどのようにトークン分割するとトークンの意味を正しく捉えられるのかは、自明ではない。

トークナイザを変更することによる日本語自然言語処理タスクの精度の違いを調査した研究 [1] は存在するが、大規模言語モデルと組み合わせた場合の精度を比較したものである。トークナイザがどれだけタスク精度に影響を与えるかについては明らかではない。

トークナイザと自然言語処理タスクの精度との関係が明らかになれば、言語処理タスク毎に適したトークナイズ手法を選ぶことや、大規模言語モデルの事前学習の初期値となるトークンベクトルを良いものにできると考えられる。

3. 比較するトークナイザ

3.1 単語分割

単語は人間にとって意味を持つ文字列の最小単位である。トークン＝単語として文を分割する手法（単語分割）は、コンピュータが人間にとって意味のある単位で特徴表現学習できる長所を持つ。しかし、単語の数（語彙数）が膨大になることは短所である。

日本語の文は、英語などと異なり、単語間に空白を挟まずに書かれる。そのため、文を単語に分割する際には形態素解析器が用いられる。形態素解析器は、語彙数が固定された単語辞書を持ち、単語の現れやすさ、つながりやすさを考慮して、文を分割する。なお、辞書に未登録の単語は未知語として扱われる。本研究では、形態素解析器として MeCab を用いる。

3.2 サブワード分割

文をサブワード分割して得られるトークン列では、トークンは単語に限定しない文字列となる。概して、サブワード分割の手法は、高頻度の 2 つの単語をつなげて 1 つのトークンとする、あるいは、低頻度の単語をより細かいトークンに分割する。その結果、サブワード分割を行うトークナイザは、トークン数（語彙数）をコントロールできる。

サブワード分割を行うトークナイザは、分割を行う際に使用する辞書（語彙集合）をコーパスから獲得するための学習過程と、文の分割を行う実行過程の 2 つの過程からなる。学習過程で獲得する辞書は、コーパス中に出現する文字を必ず含む。このため、多くの場合、文字単位まで分割することにより、未知語となるトークンを排除できる。

3.2.1 Byte-Pair Encoding (BPE)

BPE は学習で次の手順により辞書を獲得する：

① 与えられた文字列を、文字単位に分割する。
例: ABABCABCD → A / B / A / B / C / A / B / C / D
辞書[A、B、C、D]

② 隣り合う回数が最も多い 2 つを結合し、1 つのトークンとみなして辞書に加える。これを指定語彙数に達するまで繰り返しつつ、結合するルールを覚える。

例:指定語彙数が 6 の場合：
○ 1 ステップ目
A と B が隣り合う回数が最多 A / B → “AB” –(1)
A / B / A / B / C / A / B / C / D → AB / AB / C / AB / C / D
辞書[A、B、C、D、AB]
○ 2 ステップ目
AB と C が隣り合う回数が最多 AB / C → “ABC” –(2)
AB / AB / C / AB / C / D → AB / ABC / ABC / D
辞書[A、B、C、D、AB、ABC]
辞書の語彙数が 6 に達したので学習終了。

実行過程では、トークナイズしたい文章を文字単位に分割したのち、学習過程で学習したルールに則り結合する。上記の例では、(1)A+B →“AB”、(2) AB+C →“ABC”という結合ルールを学習過程で得た。この場合、実行過程では ABDABCABC という文字列は、AB / D / ABC / ABC と分割される。

3.2.2 Unigram Language Model (Unigram)

Unigram は学習過程で次の手順により辞書を獲得する：

① 設定語彙数より要素数が大きい辞書（語彙集合）を、学習用のコーパスから接尾辞配列を用いて構築する。
② 現在の辞書 V に属するトークン $x_i \in V$ の出現確率を $p(x_i)$、コーパス D に属する文を X、X に対して可能な

† 山形大学大学院理工学研究科
Yamagata University, Graduate School of Science and Engineering
‡ 岩手県立大学ソフトウェア情報学部
Iwate Prefectural University, Faculty of Software and Information Science

1) https://taku910.github.io/mecab/

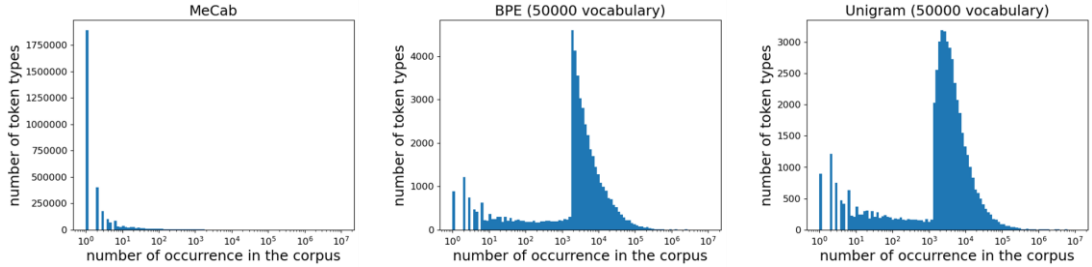

図 1 トークナイズ後のトークンの出現回数分布

サブワード分割の集合を $S(X)$、X に対して可能なサブワード分割を $x=(x_1,x_2,\ldots,x_M)$ としたとき、確率 $P(x)$を、

$$P(x)=\prod_{i=1}^{M} p(x_i)$$

とする。このとき、次の式、

$$L=\sum_{s=1}^{|D|}\log\left(P(X^{(s)})\right)=\sum_{s=1}^{|D|}\log\left(\sum_{x\in S(X^{(s)})}P(x)\right)$$

が最大になるように各トークンの出現確率 $p(x_i)$ を定める。

③ 各トークンについて、それを語彙集合に含めた状態と含めない状態とで L の差が小さいトークン、つまり、語彙集合に含めなくても影響が少ないトークンを取り除く。

③ ②と③を指定語彙数になるまで繰り返す。

実行過程では、学習過程で算出した各トークンの出現確率$p(x_i)$ を使用して、$P(x)$ が最大になるように分割を行う。

4. 実験

MeCab、BPE、Unigram の 3 つのトークナイザを用いたときの word similarity タスクの精度を比較した。

BPE と Unigram の学習には Sentencepiece [5] を用いた。学習に用いるコーパスとして、日本語 wikipedia ダンプ [2] から、コーパス中で 1 回しか出現しない文字を自明な低頻度語を作るノイズとみなし、それらを含む文を除去したものを使用した。また、BPE、Unigram で用いる辞書の語彙数は、2 万語から 10 万語まで 1 万語刻みで増加させ、9 通りの設定とした。

MeCab、および、学習した BPE、Unigram を用いて、コーパスの各文をトークンに分割し、それをもとに word2vec でトークンベクトルを作成した。トークンの類似度を、トークンベクトル同士のコサインとして計算した。

word similarity タスクの評価には日本語単語類似度・関連度データセット JWSAN[3] を使用した。JWSAN には単語のペア毎に複数名の評定者が 0〜6 の範囲で与えたスコアの平均値が示されている。単語ペアが類似する／類似しないとき、評定平均値は高い値／低い値となる。例えば、“位置”と“場所”は 4.29、“基地”と“少年”は 0.62 である。

JWSAN の評定値、および、トークンベクトル同士のコサインは、単語ペアが類似するほど値が大きくなる。このため、それらの相関係数が大きいほど、人間の感覚に近いトークンベクトルが得られている、つまり、優れたトークナイザであると考えられる。なお、相関係数の算出に用いた単語ペアは、MeCab、BPE、Unigram の語彙集合に属する 184 組の単語ペアとした。

表 1 JWSAN 評定値とコサイン類似値との相関係数

手法名	トークナイザが持つ辞書の語彙数								
	20000	30000	40000	50000	60000	70000	80000	90000	100000
BPE	0.487	0.530	0.542	0.551	0.564	0.557	0.557	0.560	0.557
Unigram	**0.502**	**0.546**	**0.561**	**0.575**	**0.570**	**0.572**	**0.562**	**0.568**	**0.566**
MeCab	0.519								

5. 結果

表 1 は、各トークナイザを用いて word similarity タスクを実施し得られた相関係数である。サブワード分割（BPE、Unigram）と単語分割（MeCab）を比較すると、語彙数 2 万語の場合を除き、サブワード分割のほうが高い相関係数を得た。また、BPE と Unigram を比較すると、Unigram のほうが高い相関係数を得たことが見て取れる。

6. 考察

サブワード分割（BPE、Unigram）が単語分割（MeCab）よりも word similarity タスクで高い精度を得た理由は、コーパスでの出現回数の少ないトークン（低頻度のトークン）が少ないからであると我々は考える。なせなら、トークンのベクトル表現は、そのトークンのコーパスでの出現をエビデンスとして構築される。したがって、コーパスでの出現回数の少ないトークンに対しては、トークンの意味を保持するトークンベクトルを得られにくいと考えられる。

MeCab、BPE、Unigram を用いてコーパスをトークナイズしたときのトークンの出現回数分布を図 1 に示す。MeCab は低頻度のトークンが非常に多いことが見て取れる。これに対して、BPE と Unigram は低頻度のトークンが大きく減少している。さらに、BPE と Unigram を比較すると、タスク精度が Unigram より劣る BPE は、中頻度トークンの出現回数分布の形が Unigram よりも尖っている。このことから、低頻度トークンの数以外にも word similarity タスクの精度に影響を及ぼす要因が存在すると考えられる。

トークンの出現回数分布に焦点を当て、タスク精度に影響する条件の調査を今後の研究課題としたい。

参考文献

[1] 藤井, 柴田, 山口,他, “日本語 Tokenizer の違いは下流タスク性能に影響を与えるか？”, 言語処理学会第 29 回年次大会, 2023.
[2] Philip Gage, “A new algorithm for data compression”, C Users J. 12(2):23–38, 1994.
[3] Rico Sennrich, Barry Haddow, and Alexandra Birch, “Neural machine translation of rare words with subword units”, arXiv, 2015.
[4] Taku Kudo, “Subword regularization: Improving neural network translation models with multiple subword candidates”, arXiv, 2018.
[5] Taku Kudo, John Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing”, In Proc. EMNLP, 2018.
[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space”, In Proc. ICLR, 2013.

2) https://dumps.wikimedia.org/jawiki/latest/jawiki-latest-pages-articles.xml.bz2
3) http://www.utm.inf.uec.ac.jp/JWSAN/jwsan-1400.csv