Vol.2018-NL-236 No.2
2018/7/9

教師なしニューラル単語分割を用いた分散表現獲得

平岡 達也[1,a)]　進藤 裕之[1]　松本 裕治[1]

概要：教師なし単語分割はギブスサンプリングによる言語モデルの尤度最大化として行う手法が提案されている [持橋,2009]．また，分割の際に用いる言語モデルをニューラルネットワークに置き換える試みも [友利,2018] にて提案されている．しかしニューラルネットワークを用いた手法には計算コストが高いという問題がある．本論文ではモデルの単純化により計算コストを軽減した新たな手法について提案する．さらに分割によって得られた単語分散表現が，日本語の文分類タスクにおいて有効であることを確認した．

キーワード：教師なし単語分割，ニューラルネットワーク，言語モデル，Forward Sampling-Backward Filtering，ギブスサンプリング

Word Representation Learning with Unsupervised Neural Word Segmentation

Tatsuya Hiraoka[1,a)]　Hiroyuki Shindo[1]　Yuji Matsumoto[1]

1. 背景

日本語や中国語のように分かち書きを行わない言語では，テキストから情報抽出や検索を行う際に，前処理として単語分割を必要とすることが多い．従来の単語分割手法として，MeCab[1] や Byte Pair Encoding (BPE)[2] が挙げられる．しかし，人手で整備した単語辞書を用いる MeCab では未知語やインフォーマルな表現を多く含む文に対して分割精度が低下してしまい，頻度順にサブワードを生成する BPE では文脈を考慮した分割ができないという問題がある．

このような背景のもと，持橋らによって教師なし学習による単語分割の手法が提案されている [3]．持橋らの手法では，N-gram 言語モデルの尤度を最大化する分割が最適であるという基準を用いており，単語辞書を用いずに文脈を考慮した分割が得られるという利点がある．

さらに近年では，分割に用いる言語モデルをリカレントニューラルネットワーク（RNN）に置き換えた手法が提案されている [4]．しかしながら，ニューラル言語モデルは語彙の増大とともに計算コストが高くなり，学習の収束に時間がかかることや，大規模なデータに適用できないという問題がある．

本研究はこれらの問題点を解決するために，ニューラル言語モデルを用いた新たな教師なし単語分割モデルの提案を行う．また，提案手法によって得られる分割および分散表現が，実際の自然言語処理タスクにおいて有効であることを日本語 Twitter 評判分析データセット [5] および NTCIR6 意見分析パイロットタスク [6] について確認した．

2. 先行研究

2.1 教師なし単語分割

教師なし単語分割は，ある文章 D に含まれる文字列 $s = c_1c_2...c_N$ に対して，単語列 $\mathbf{w} = w_1w_2...w_M$ が与えられる確率 $p(\mathbf{w}|s)$ を最大化するような単語列 $\mathrm{argmax}_{\hat{\mathbf{w}}} p(\hat{\mathbf{w}}|s)$ を求める問題として定式化することができる．分割による単語列の確率 $p(\mathbf{w}|s)$ は言語モデルを用いた文の尤度の計算によって与えることができ，この言語モデルの尤度最大化を目指すことで，文字列 s に対する最適な単語列 $\hat{\mathbf{w}}$ が得られる．

教師なし単語分割においては，データ D を構成する語彙

[1] 奈良先端科学技術大学院大学
NARA Institute of Science and Technology
[a)] hiraoka.tatsuya.hq7@is.naist.jp

V_D そのものが未知である．そのため，文字列 s を分割することによって得られる，あらゆる単語系列 $\mathbf{w}$ に対して確率 $p(\mathbf{w})$ を計算できる必要がある．例えば，離散バイグラムを用いた言語モデルの場合，$p(\mathbf{w})$ は以下のように計算される．

$$p(\mathbf{w}) = p(w_1|<\text{BOS}>)p(<\text{EOS}>|w_M)\prod_{m=2}^{M} p(w_m|w_{m-1}) \tag{1}$$

<BOS>と<EOS>はそれぞれ文頭，文末を表すトークンである．あらゆる単語 w_m に確率を与えられるようなバイグラム確率 $p(w_m|w_{m-1})$ は以下の式によって与えられる．

$$p(w_m|w_{m-1}) = \frac{\text{count}_D(w_m|w_{m-1}) + \alpha p_{base}(w_m)}{\text{count}_D(w_{m-1}) + \alpha} \tag{2}$$

ここで，$\text{count}_D(w)$ は，単語分割が行われた文章 D における単語 w の頻度であり，$\text{count}_D(w_m|w_{m-1})$ は，単語 w_{m-1} の直後に w_m が出現するような文字列の頻度である．また，$p_{base}(w)$ は，w の基底確率である．$p_{base}(w)$ は，一様分布または文字 n-gram などを用いて計算され，スムージングの役割を果たす．また，α は基底分布 $p_{base}(w)$ の重みである．

(1) 式の尤度が最大となる単語列 $\hat{\mathbf{w}}$ を求める方法として，ギブスサンプリングを用いる手法が提案されている [3]．いま，ある文 s が $\mathbf{w}_s$ という単語列に分割されているとする．このとき，$\mathbf{w}_s$ に含まれる各単語のカウントを言語モデルから削除したうえで，s の情報を含まない言語モデルを用いて新たな分割 $\mathbf{w}_s$ をサンプリングし，言語モデルに追加する．これを全ての文に対して繰り返し行うことで，言語モデルの尤度が高くなる分割のサンプルを得ることができる．

文献 [3] では，Forward filtering-Backward sampling (FFBS) と呼ばれる動的計画法の一種を用いることで，単語ごとではなく，文ごとに単語列をサンプリングしている．

2.2 ニューラル言語モデルを用いた分割

言語モデルの尤度最大化に基づく単語分割では，言語モデル自体の性能が分割結果に大きく影響する．友利らは，分割に用いる言語モデルをニューラルネットワークに置き換えた手法を提案している [4]．彼らの手法では，文脈 h から単語 w が生成される確率 $p(w|h)$ を，以下のように定義する．

$$p(w|h) = (1-\lambda)p_w(w|h) + \lambda p_c(w) \tag{3}$$

これは，単語のニューラル言語モデル $p_w(w|h)$ と，文字のニューラル言語モデル $p_c(w)$ が係数 λ によって重みづけされた形になっている．p_w，p_c として RNN 言語モデルが採用されている．

ニューラル言語モデルで単語確率を計算する際，語彙 V_D に対応する単語分散表現の行列 $\boldsymbol{V}_D$ を用いるが，教師なし単語分割では V_D が変動する．文献 [4] では現在分割されている D における単語の頻度をカウントすることで，語彙 V_D を管理し，$\text{count}_D(w) \geq 1$ となる単語 w については単語分散表現 $v_w \in \boldsymbol{V}_D$ を保持する．語彙に含まれる単語 $\{w \in V_D|\text{count}_D(w) \geq 1\}$ については，p_w から単語確率を計算し，語彙に含まれない単語については計算を行わない．この時，p_c があらゆる単語に対して確率を与えるため，言語モデル全体で確率が 0 となる単語は存在しない．また (3) 式の尤度が最大となる単語列を求めるために，NPYLM[3] と同様にギブスサンプリングが用いられている．

文献 [4] で用いられている RNN 言語モデルは，過去の全ての文脈に出現した単語に依存して次の単語の確率が決定される．そのため，動的計画法である FFBS を用いた文ごとのサンプリングができない．そこで文献 [4] では，RNN 言語モデルによる $t-1$ 単語目までの文脈 $\mathbf{w}_{1:(t-1)}$ から単語 w_t が生成される確率 $p(w_t|\mathbf{w}_{1:(t-1)})$，および w_t を選んだ時の報酬スコア r_t の積 $p(w_t|\mathbf{w}_{1:(t-1)})r_t$ を用いて単語ごとに前からサンプリングを行う．具体的には，r_t は次の式によって定義される．

$$r_t = \frac{1}{Z}\sum_{z=1}^{Z} LM(\mathbf{w}^z_{1:T}), \mathbf{w}^z_{1:T} \in \text{MC}^{\theta}(\mathbf{w}_{1:t}, s; Z) \tag{4}$$

$$LM(\mathbf{w}_{1:T}) = \frac{1}{T}\sum_{t=1}^{T} p(w_t|\mathbf{w}_{1:(t-1)}) \tag{5}$$

報酬 r_t は単語 w_t を選んだ場合に，その選択が後ろの分割にどの程度影響するかを表したスコアである．t 番目の単語 w_t を選んだとき，文 s について可能な分割 $\mathbf{w}$ をパラメータ θ によるモンテカルロ探索を用いて Z 回サンプリングする．得られた Z 個の分割 $\mathbf{w}^z$ のスコアはそれぞれ (5) によって計算され，これらの平均をとることで $p(\mathbf{w}_{(t+1):T})$ についての周辺化を近似している．

2.3 問題点

文献 [4] によるニューラル言語モデルを用いた分割手法には，以下のような問題点がある．

(a) 計算コストが高い

単語ごとのギブスサンプリングを用いた手法では，単語が生成される確率や報酬 r_t を何度も計算する必要があるが，離散的な言語モデルと比較して RNN 言語モデルは計算コストが高いため，大量のデータを学習に使うことが難しい．

(b) 局所解に陥る可能性がある

文献 [4] では，単語のサンプリング時，Z 個の分割に対

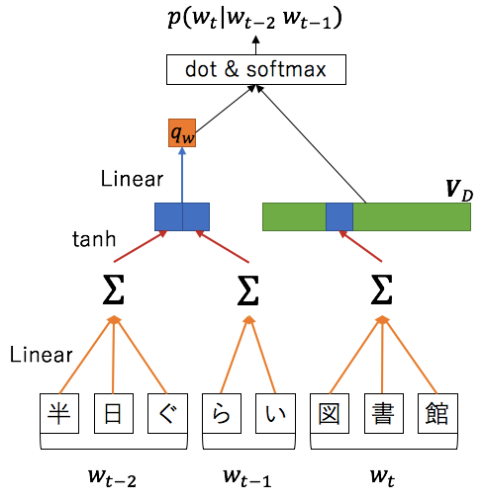

図 1 単語言語モデル p_w の計算概要

する言語モデルの尤度期待値を用いて報酬 r_t を計算する．しかし，計算コストの高さから，学習時に $Z = 1$ としているため，局所的な高スコアに影響されやすい．さらに，報酬 r_t は確率の平均から導出されるため，語彙が小さいほどスコアが高くなる傾向にある．そのため，言語モデルによる文の尤度を反映した分布から単語をサンプリングすることができない．

3. 提案手法

上記の問題を緩和するために，我々は RNN 言語モデルの代わりに，ニューラルネットワークに基づくトライグラム言語モデルを用いる．ニューラルトライグラム言語モデルでは，単語の生成確率が前の 2 単語にしか依存しないため，動的計画法に基づく文レベルのギブスサンプリング（FFBS）が実現できる．その一方で，従来の離散的なトライグラム言語モデルと比べて，単語の分散表現を用いることによりデータスパースネスの問題を緩和できるという利点がある．また，ニューラルネットワークに基づく言語モデルでは，出力層に用いる softmax 関数の計算コストが高いという問題がある．そこで我々は，語彙をサンプリングすることにより，softmax 関数の計算コストを抑えることを試みた．

3.1 言語モデル

3.1.1 単語レベルの言語モデル

提案手法では，文献 [4] と同様に，単語レベルの言語モデル p_w と，文字レベルの言語モデル p_c を係数 λ で重みづけした言語モデルを用いて単語分割を行う．単語レベルの言語モデル p_w として，ニューラルトライグラム言語モデルを用いる．これは，w_t の生成確率を直前の 2 単語 $w_{t-2}w_{t-1}$ から計算する言語モデルであり，以下のように表される．

$$p_w(w_t|w_{t-2}w_{t-1}) = \mathrm{softmax}(q_w^\top \boldsymbol{V}_D)_{w_t} \tag{6}$$

$$q_w = \tanh(\boldsymbol{W}_{q_w}(v_{w_{t-2}}; v_{w_{t-1}}) + b_{q_w}) \tag{7}$$

ここで，$\boldsymbol{V}_D$ は，現在のモデルにより単語分割を行った文章 D に含まれる単語集合である．また，v_w は，単語 $w \in \boldsymbol{V}_D$ に対応する分散表現ベクトルである．現在の語彙に含まれないトライグラムについては，単語レベルの言語モデル p_w の計算を行わずに確率を 0 とする．

単語を文字の系列とみなし，$w = c_1...c_J$ とすると，単語 w の分散表現 v_w は，文字レベルの分散表現 $u_{c_1}, ..., u_{c_J} \in \boldsymbol{U}_D$ を用いて，以下の式から計算する．

$$v_w = \tanh(\sum_{j=1}^{J} \boldsymbol{W}_j u_{c_j} + b_j) \tag{8}$$

$\boldsymbol{W}_j$ と b_j はそれぞれ，j 文字目に対応するパラメータである．このように，文字位置ごとにパラメータを用意して，そのパラメータと文字分散表現を合成することによって単語の分散表現 v_w を得る．これは，Cai らによる分散表現の計算 [8] を応用したものであり，単語の最大長 L を固定することで高速に文字分散表現から単語分散表現を得ることができるという利点がある．

文頭および文末は<BOS>，<EOS>を一文字の単語とみなしたトークンによってパディングする．この単語言語モデル p_w の計算概要を図 1 に示した．

3.1.2 文字レベルの言語モデル

次に，文字レベルの言語モデルについて説明する．(3) 式における単語確率 $p_c(w)$ は，文字レベルのニューラルバイグラム確率 p_n によって計算する．すなわち，J 文字長の単語 $w = c_1...c_J$ の確率 $p_c(c_1...c_J)$ は，

$$p_c(c_1...c_J) = p_d(c_1)\prod_{j=2}^{J} p_n(c_j|c_{j-1}) \tag{9}$$

となる．ただし，例外的に 1 文字目の確率は離散ユニグラム確率 p_d を用いており，入力文章 D から事前に計算する．これはサンプリングによって文字レベルの言語モデルから計算される単語確率が大きく偏るのを防ぐためである．具体的な p_d，p_n の式は，以下の通りである．

$$p_d(c_j) = \frac{\mathrm{count}_D(c_j)}{\sum_{c' \in D} \mathrm{count}_D(c')} \tag{10}$$

$$p_n(c_j|c_{j-1}) = \mathrm{softmax}(q_c^\top \boldsymbol{U}_D)_{c_j} \tag{11}$$

$$q_c = \boldsymbol{W}_{q_c} u_{c_{j-1}} + b_{q_c} \tag{12}$$

文字レベルの言語モデル p_c は，サンプリングされた単語ごとに別々に学習を行うため，単語境界を跨いだ文字の連結は学習されない．

3.1.3 言語モデルの統合

最後に，文献 [4] と同様に，単語レベルの言語モデル p_w と文字レベルの言語モデル p_c を係数 λ によって統合する．

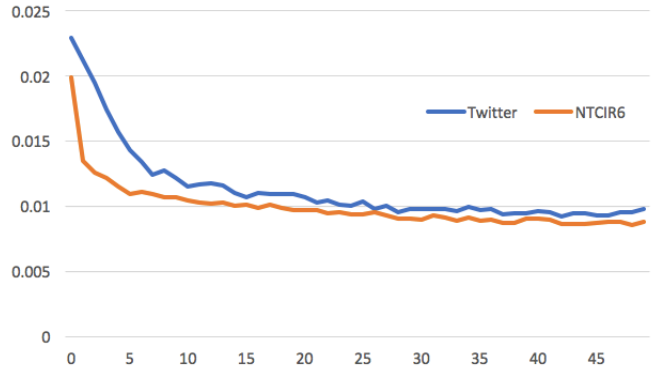

図 2　λ の推移:横軸は学習エポック

(3) 式における λ は，文字言語モデル p_c の影響力の強さを表している．すなわち，λ が大きいほど単語言語モデル p_w の影響が小さくなり，既知語の確率 $p(w_{IV} \in V_D)$ と未知語の確率 $p(w_{OOV} \notin V_D)$ との差が小さくなる．

適切な単語分割を得るためには，分割初期に λ を大きめに取り，分割が収束するにつれて小さく設定することが望ましい．本研究では全単語数に占める低頻度語の割合を用いて以下のように λ を設定する．

$$\lambda = \frac{|\{w \in V_D | \mathrm{count}_D(w) = 1\}|}{\sum_{w' \in V} \mathrm{count}_D(w')} \tag{13}$$

ここで低頻度語とは，$\mathrm{count}_D(w) = 1$ となるような w とする．このような単語 w は単語カウントの削除によって，サンプリング時に毎回未知語となるためである．後述する実験データに対する λ の推移を図 2 に示した．学習が進むにつれて分割が収束に向かい，低頻度語の割合が減少していることが確認される．

3.2　学習方法

学習イテレーションは先行研究 [3] [4] と同様に，モデルから単語の削除，サンプリング，追加を繰り返すギブスサンプリングによって行われる．学習イテレーションを図 3 に示した．

図 3 において，remove と add は言語モデル (LM) の単語カウントの削除と追加，update はニューラル言語モデルの更新を表す．提案手法はフィードフォーワード型のニューラルネットワークを用いて文レベルでサンプリングを行うため，[4] に比べて計算コストが低く，サンプリングの効率も高い．実際に，友利らの手法では，既存の教師なし単語分割手法によってあらかじめ単語分割を行い，それを初期値として学習を始めているが，我々の提案手法では，ランダムな初期分割状態からスタートしても，十分に事後確率の高い分割結果を得ることができる．

また，提案手法では，複数文をミニバッチとしてまとめ，ミニバッチ単位で学習を行う．そして，通常の言語モデルと同様に，単語レベルの言語モデル p_w は文ごと，文字レベルの言語モデル p_c は単語ごとの交差エントロピー誤差によって学習を行う．

```
set LM with random segmentation
while j < max_epoch do
    for each mini_batch do
        remove w ∈ s_mini_batch from LM
        sample k-words
        sample new s_mini_batch
        add w ∈ s_mini_batch to LM
        update LM with s_mini_batch
    end for
end while
```

図 3　学習イテレーション

3.3　ソフトマックス関数の近似

(6) 式で示されるようなニューラルネットワークに基づく言語モデルでは，出力層にソフトマックス関数を用いて単語確率を計算する必要があり，これが計算速度のボトルネックとなっている．そこで我々は，学習時に限り，以下のような近似を用いた高速化を行う．

$$\begin{aligned} p(w|h) &= \frac{\exp(q^\top v_w)}{\sum_{w'} \exp(q^\top v_{w'})} \\ &\simeq \frac{\exp(q^\top v_w)}{\frac{|V_D|}{K} \sum\limits_{\substack{w'_1 \ldots w'_K \\ \sim \mathrm{uniform}(V_D)}} \exp(q^\top v_{w'_k})} \end{aligned} \tag{14}$$

つまり，現在の分割に含まれる語彙 V_D から K 単語をサンプリングし，これを使ってソフトマックス関数の分母を近似する．実際には，$K = 100$ 程度を用いる．

なお，文字言語モデル p_c では文字種類数が固定であるため，近似による高速化を必要としない．

4.　実験

提案手法の有効性を確認するため，日本語テキストに対して教師なし単語分割を行なった．ただし，教師なし学習による単語分割の結果を評価することは難しい．そこで，提案手法によって得られた単語分散表現を文分類タスクの学習に用い，その効果について評価を行い，提案手法の有効性を評価する．また，提案手法の Chainer による実装と，使用したデータに関する情報は GitHub で公開している[*1]．

4.1　実験データ

日本語のデータとして，Twitter 日本語評判分析データセット [5] および NTCIR6 意見分析パイロットタスク [6] を用いた．Twitter データセットは，ある商品名を含む Twitter 上の投稿が，商品に対してどのような立場による

[*1] https://github.com/tatHi/neural_uws

表 1　データ内訳:ラベルは順に positive and negative, positive negative, neutral, unrelated を表す．単位は文．

		P/N	POS	NEG	NEU	U/R	
Twitter	訓練	38	482	889	6737	1852	9998
	評価	7	96	177	1347	370	1997
NTCIR6	訓練	-	1540	3063	1904	-	6507
	評価	-	111	417	315	-	843

ものかについてラベル付けされた文分類タスクである．今回用いるデータは，その中で iPhone に関するツイートに限定したものである．

NTCIR6 については，毎日新聞記事へアノテーションされたデータのうち，極性ラベルが付与された文のみを用いる．ラベルが複数付与された文については，文全体の極性を正解ラベルとし，トピック 1-29 を訓練データ，30-32 を評価データとして用いる．なお，評価データのうち訓練データで未知となる文字を含む 99 文をあらかじめ除外した．

各データセットの内訳は表 1 に示した通りである．データに含まれる文字数は，Twitter が 490,202(訓練)/98,847(評価)，NTCIR6 が 309,000(訓練)/32,870(評価) である．

4.2　実験方法

教師なし単語分割器は訓練データに対してのみ分割および言語モデルの学習を行う．訓練データの分割がある程度収束したパラメータを用いて，訓練データおよび評価データのテキストを前向きアルゴリズムによって尤度最大となるように分割する．単語分割器から得られた分割と単語分散表現を用いて，文分類器の学習を行い評価する．この時，分散表現は固定とする．

比較対象として形態素解析器 MeCab[1] による分割と，BPE[2] による分割を用いる．BPE による分割は，MeCab での分割による語彙数と同程度になるよう設定した．それぞれの手法を用いて訓練データ，評価データの単語分割を行い，この単語分割結果を用いて分類器を訓練・評価する．

提案手法と他手法を同じ条件で比較するため，単語分散表現はニューラル単語トライグラム言語モデル p_w を用いて訓練データについて事前学習を行い，タスク学習時には固定する．また，事前学習に用いる単語分散表現は (8) 式を用いて，教師なし分割器と共通の文字分散表現から生成する．文字分散表現は事前に文字 CBOW によって学習済のものを用い，単語トライグラム言語モデルの学習時には固定する．実験に用いたハイパーパラメータを表 2 にまとめた．

4.3　文の分類器モデル

文の分類器は，文の分散表現 S の線形変換を用いて n 値分類を行う単純なものとした．文の分散表現を得る方法として，Deep Average Network(DAN)[9] と Long-Short Term Memory(LSTM)[10] の二種類を用意した．文の分類モデル，LSTM による文の分散表現のモデル，DAN による文の分散表現のモデルは，それぞれ以下の式で示される．

$$p(y = l|S) = \mathrm{softmax}(\boldsymbol{W}_c S + b_c)_l \tag{15}$$

$$\begin{aligned} S_{\mathrm{LSTM}} &= o_M \\ o_m &= \mathrm{LSTM}(o_{m-1}, v_{w_m}) \end{aligned} \tag{16}$$

$$S_{\mathrm{DAN}} = \boldsymbol{W}_{\mathrm{DAN}}(\frac{1}{M}\sum_{w \in \mathbf{w}} v_w) + b_{\mathrm{DAN}} \tag{17}$$

ただし，M は単語数で，v_w は事前学習された単語分散表現である．LSTM は前向き一層の最終隠れ状態 o_M を文分散表現 S_{LSTM} とする．DAN は文に含まれる単語分散表現の平均の線形変換を S_{DAN} とする．

DAN による S_{DAN} の計算ではあらゆる単語の分散表現の平均に基づくモデルであるため，比較する 3 種類の単語分割結果に基づく分散表現を全て足し合わせたモデルについても実験を行う．これらの文の分類器の学習は，正解ラベルに対する交差エントロピー誤差を用いて行った．

表 2　ハイパーパラメータ

char-embedding size	30
word-representation size	100
最大単語長 L	8
文分散表現 S	100
サンプリングサイズ K	100
分割器ミニバッチサイズ	32
BPE 単語数 (Twitter)	226,994(訓練), 45,862(評価)
BPE 単語数 (NTCIR6)	186,345(訓練), 19,581（評価）
文分類器学習 epoch 数	100
文分類器ミニバッチサイズ	64
LSTM dropout	0.2
DAN word-dropout	0.2

表 3　分割による性能比較 (F 値)

	Twitter		NTCIR6	
	DAN	LSTM	DAN	LSTM
BPE	0.7044	**0.7057**	0.4818	0.4744
MeCab	0.7037	0.6978	0.4791	0.4673
提案手法	**0.7046**	0.7034	**0.4834**	**0.4830**
BPE+MeCab	0.7039	-	0.4833	-
BPE+提案手法	0.7042	-	0.4819	-
MeCab+提案手法	0.7040	-	**0.4866**	-
BPE+MeCab+提案手法	**0.7048**	-	0.4814	-

4.4　結果と考察

それぞれの分類器の学習を 100 エポック行い，F 値の平均を表 3 に示した．Twitter データセットにおいては，辞書を用いる MeCab が最も精度が低く，ついで提案，BPE の順である．一方，NTCIR6 においては提案手法が最も精度が良く，有効性が確認された．また，DAN において複数

表 4　手法による分割比較

データ	手法	分割
Twitter(訓練)	BPE	半｜日｜ぐらい｜図｜書｜館｜に｜いて｜お｜も｜た｜。｜ iPhone6S ｜め｜っ｜さ｜も｜つ｜やん
	MeCab	半日｜ぐらい｜図書｜館｜に｜いて｜おもた｜。｜ iPhone ｜ 6 ｜ S ｜め｜っさ｜も｜つや｜ん
	提案	半日｜ぐらい｜図｜書館｜に｜いて｜お｜もた｜。｜ iPho ｜ ne ｜ 6S ｜めっ｜さ｜もつ｜やん
Twitter(評価)	BPE	2 ｜日｜充電｜せん｜でも｜い｜けた｜ iPhone6S ｜ plus ｜やば｜いな
	MeCab	2 ｜日｜充電｜せん｜でも｜いけた｜ iPhone ｜ 6 ｜ Splus ｜や｜ばい｜な
	提案	2日｜充電｜せん｜でも｜いけ｜た｜ iPho ｜ ne ｜ 6S ｜ plus ｜やば｜いな
NTCIR6(訓練)	BPE	喫｜煙｜論｜争｜が｜なか｜なか｜決｜着｜しない｜四｜つ｜の問題｜点｜を｜指摘｜した｜い。
	MeCab	喫煙｜論争｜が｜なかなか｜決着｜し｜ない｜四つ｜の｜問題｜点｜を｜指摘｜し｜たい｜。
	提案	喫煙｜論争｜が｜なかなか｜決｜着｜しない｜四つ｜の｜問題点｜を｜指摘｜したい｜。
NTCIR6(評価)	BPE	第｜二｜に｜基礎｜学力｜の｜重｜視｜にも｜半｜歩｜前｜進｜が｜見｜られ｜る。
	MeCab	第｜二｜に｜基礎｜学力｜の｜重視｜に｜も｜半｜歩｜前進｜が｜見｜られる｜。
	提案	第二｜に｜基礎｜学｜力の｜重視｜にも｜半｜歩｜前進｜が｜見られる｜。

のモデルの分割結果を統合する実験では，いずれも提案手法による分割結果を加えることにより，性能向上が見られた．したがって，提案手法による教師なし単語分割は，文分類タスクにおいて一定の性能向上に寄与するといえる．

各分割器によって得られた訓練データ，評価データに対する実際の分割結果は表 4 の通りである．Twitter データセットについては，「やばい」や「もつやん」といった崩れた表現において，提案手法が MeCab や BPE よりも適切な分割であることが確認できた．一方で「iPho ｜ ne」といった不適切な分割も見られる．これは「iPhone」と「iphone」という表記の揺れが影響していると考えられる．Twitter データセットでは訓練データと評価データの性質が同じであり語彙の規模も大きくないため，評価データに対して安定した分割を得る事ができた．

NTCIR6 は新聞記事を用いているため語彙の規模が大きく，辞書ベースの MeCab が威力を発揮した．BPE や提案手法は低頻度語が多いテキストには弱いため，特に評価データにおいて適切な分割が得られない事がある．また，NTCIR6 では訓練データと評価データのトピックが異なるため，訓練データには含まれない語彙が評価データに多く出現し，分割の失敗が生じていた．提案手法では計算の高速化のために単語トライグラムの確率を単語バイグラム確率でスムージングする計算を省略しており，これが低頻度語の分割ミスに繋がっていると考えられる．

表 5　提案手法と先行研究 (執筆者実装) の速度比較

友利ら*2[4]	11.56 秒/1 文
提案	**0.28 秒/1 文**

次に，提案手法と先行研究 [4] の分割速度*3について表 5 にまとめた．提案手法は，先行研究と比較して，大きな速度改善が見られる．また，提案手法において，50 エポックへの到達時間は Twitter データで 2,325 分，NTCIR で 1,218 分であった．今回の実験設定では，エポック数を 100 としていることから，実データにおいて，提案手法は現実的な速度で単語分割の結果を得られることがわかる．

*2　先行研究 [4] は計算に多大な時間を要するため，1 エポックのみの計測とした．
　また，文脈と単語のサンプリングには MeCab による正解を与え，収束状態に近い環境での計測を行った．実際の分割では未知語が頻出するため，さらに時間がかかると考えられる．

*3　Twitter データセットの訓練データ 9,998 文について，18 コア Xeon E5-2699v3 (RAM:512GB) を用いた．

5. まとめ

本論文では，ニューラルトライグラム言語モデルと，文レベルのサンプリングを用いた教師なし単語分割手法を提案した．また，提案手法による単語分割の結果から得られる分散表現を用いて日本語の文分類タスクの実験を行い，特にインフォーマルな文章において提案手法による分散表現獲得が有効であることを示した．

今後は，日本語に限らず中国語や英語のタスクについても本手法の分割が有効であるかを確認していきたい．

参考文献

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

[2] 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)*, Vol. 1, pp. 1715–1725, 2016.

[3] Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pp. 100–108. Association for Computational Linguistics, 2009.

[4] 友利涼, 森信介. ニューラルネットワークによる教師なし単語分割. 言語処理学会第 24 回年次大会発表論文集 (NLP2018), 2018.

[5] Yu Suzuki. Twitter 日本語評判分析データセット. *http://bigdata.naist.jp/ ysuzuki/data/twitter/*, 2017.

[6] Yohei Seki, David Kirk Evans, Lun-Wei Ku, Hsin-Hsi

Chen, Noriko Kando, and Chin-Yew Lin. Overview of opinion analysis pilot task at ntcir-6. In *NTCIR*, 2007.

[7] Yee Whye Teh. A hierarchical bayesian language model based on pitman-yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 985–992. Association for Computational Linguistics, 2006.

[8] Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. Fast and accurate neural word segmentation for chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vol. 2, pp. 608–615, 2017.

[9] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. 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)*, Vol. 1, pp. 1681–1691, 2015.

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