確率的トークナイザーを用いたニューラル感情分析

平岡 達也　　進藤 裕之　　松本 裕治
奈良先端科学技術大学院大学
{hiraoka.tatsuya.hq7, shindo, matsu}@is.naist.jp

1 はじめに

単語分割は感情分析などの文分類タスクにおいて重要な問題である．ニューラルネットワークを用いた文分類では，文をトークンに分割した上で文ベクトルを計算する．具体的には，文は文字や単語，サブワードといったトークンに分割され，トークンに対応するベクトルを LSTM などを用いて文ベクトルへと変換する．

日本語や中国語など，スペースによって単語境界を明示しない言語においては，単語分割はさらに重要な問題となる．こうした言語においては，MeCab のような辞書ベースのツールや，ニューラルネットワークによる単語分割が行われている．しかし，これらの手法には人手で整備された辞書や，単語分割に関するアノテーションが大量に必要となるという問題がある．

本研究では，教師なし単語分割の手法を用いた確率的な単語分割モデルについて提案する．提案手法は言語モデルを用いて学習データを確率的に分割し，さらにサンプリングされた分割をテスト時に利用できるように言語モデルを更新する．

近年では機械翻訳の分野においてサブワード正則化 [4] が提案されており，サブワード候補をサンプリングすることによる精度向上が報告されている．本研究はこの流れを汲んでおり，(a) 語彙サイズを固定しない (b) サンプリングされた分割を用いて言語モデルを更新するという二点において異なっている．日本語，中国語，英語による感情分析タスクにおいて実験を行い，提案手法が分類器の精度を向上させることを確認した．

2 ニューラル文分類器

本研究では，LSTM によるニューラル文分類器を用いる．分類器はある文 s を入力とし，文に付与されたラベルの予測値 $\boldsymbol{y}$ を出力する．文 s が入力された時，分割器を用いてこれを N 個のトークン $t_1...t_n...t_N$ に分割する．あるトークン t の分散表現は，Lookup テーブル $\boldsymbol{V}^{\text{token}}$ を用いて以下のように割り当てられる．

$$\boldsymbol{v}_t = \boldsymbol{V}^{\text{token}}\boldsymbol{k}_t \tag{1}$$

ここで $\boldsymbol{k}_t$ は t に対応する要素が 1 となる one-hot 表現である．

Lookup テーブルを用いた単語埋め込みは，語彙の規模と同じだけ分散表現を保持しなければいけない．また，テストデータに含まれる未知語は全て同一の未知語を表す分散表現として表されるという問題点がある．これらの問題は，文字レベルの分散表現を用いてトークンベクトル $\boldsymbol{v}_t$ を計算することで解決できる．トークン t が M 文字 $c_1...c_m...c_M$ から構成されている時，$\boldsymbol{v}_t$ は一層の単方向 LSTM を用いて以下のように計算される．

$$\begin{aligned}\boldsymbol{v}_t &= \boldsymbol{o}_M^{\text{char}} \\ \boldsymbol{h}_m^{\text{char}}, \boldsymbol{o}_m^{\text{char}} &= \text{LSTM}(\boldsymbol{h}_{m-1}^{\text{char}}, \boldsymbol{o}_{m-1}^{\text{char}}, \boldsymbol{v}_{c_m})\end{aligned} \tag{2}$$

$$\boldsymbol{v}_{c_m} = \boldsymbol{V}^{\text{char}}\boldsymbol{k}_{c_m} \tag{3}$$

LSTM(·) は直前の状態 $\boldsymbol{h}_{m-1}^{\text{char}}$ と出力 $\boldsymbol{o}_{m-1}^{\text{char}}$，および現在の入力 $\boldsymbol{v}_{c_m}$ を用いて，現在の状態と出力を計算する LSTM 機構である．$\boldsymbol{v}_{c_m}$ は文字レベルの分散表現で，(1) と同様に Lookup テーブルを用いて計算される．(2) によって計算された最後の出力 $\boldsymbol{o}_M$ を，トークンの分散表現 $\boldsymbol{v}_t$ として用いる．

トークンレベルの分散表現と，文字情報を用いて計算した分散表現の両方を利用することで，タスクの精度向上が期待される．文字およびトークンレベル双方の情報を用いたトークンベクトル $\boldsymbol{v}_t$ は，以下のようにベクトル結合とアフィン写像を用いて計算される．

$$\boldsymbol{v}_t = \boldsymbol{W}^{\text{cat}}(\boldsymbol{v}_t^{\text{token}}; \boldsymbol{v}_t^{\text{char}}) + \boldsymbol{b}^{\text{cat}} \tag{4}$$

ここで $\boldsymbol{v}^{\text{token}}$ と $\boldsymbol{v}^{\text{char}}$ はそれぞれ (1) と (2) を用いて計算されたトークンの分散表現を表す．また，$\boldsymbol{W}^{\text{cat}}$ と $\boldsymbol{b}^{\text{cat}}$ は学習を行うパラメーターである．

文 s の分散表現 $\boldsymbol{v}_s$ は単方向 LSTM を用いて以下のように $\boldsymbol{v}_t$ から計算される．

$$\begin{aligned} \boldsymbol{v}_s &= \boldsymbol{o}_N^{\text{token}} \\ \boldsymbol{h}_n^{\text{token}}, \boldsymbol{o}_n^{\text{token}} &= \text{L}STM(\boldsymbol{h}_{n-1}^{\text{token}}, \boldsymbol{o}_{n-1}^{\text{token}}, \boldsymbol{v}_{t_n}) \end{aligned} \tag{5}$$

(2) と同様，最後のトークンベクトル $\boldsymbol{v}_{t_N}$ を入力した際の出力を文ベクトルとして用いる．

文 s のラベルが u である確率 $p(y_s = u \mid \boldsymbol{v}_s)$ は，文ベクトルを用いて次のように計算される．

$$p(y_s = u \mid \boldsymbol{v}_s) = \text{softmax}(\boldsymbol{W}^{\text{dec}}\boldsymbol{v}_s + \boldsymbol{b}^{\text{dec}})_u \tag{6}$$

$\boldsymbol{W}^{\text{dec}}$ と $\boldsymbol{b}^{\text{dec}}$ はアフィン写像のパラメータ，$\text{softmax}(\cdot)_u$ はソフトマックス関数を適用しベクトルの u 番目の要素を抜き出す操作である．分類器は予測した y_s と正解ラベルとの交差エントロピー誤差の最小化により学習される．

3 提案手法

3.1 概要

本手法はニューラル文分類器に入力する文を確率的に分割するモデルである．学習時に言語モデルを用いて文の分割をサンプリングし，テスト時には直近のサンプリング結果を考慮して文を分割する．例えば学習時に「東海道新幹線」というフレーズを「東海道／新幹／線」と分割した時，テストデータに含まれる類似した表現「東北新幹線」は「東北／新幹線」ではなく「東北／新幹／線」と分割することで学習の内容が効率よく発揮されると考えられる．

具体的な処理手順として，学習時に文が入力された時，本手法はまず言語モデルに基づいて文を確率的に分割する．得られた分割を用いて言語モデルを更新した上で，分割後の文を後段の分類器へと入力する．テスト時には確率的な分割を行わず，ビタビアルゴリズムを用いて文の尤度が最大となるような分割をおこなう．この時，学習時にサンプリングされた分割で言語モデルを更新しておくことで，直近に入力された分割を考慮して文の分割が可能となる．

3.2 言語モデル

本手法では分割をサンプリングするために，教師なし単語分割において提案されているユニグラム言語モデル [2] を用いる．トークン t が M 文字 $c_1 ... c_M$ から構成される時，トークンのユニグラム確率 $p(t)$ は以下のように計算される．

$$p(t) = \frac{\text{count}(t) + \alpha p_{\text{base}}(t)}{\sum_{\hat{t}} \text{count}(\hat{t}) + \alpha} \tag{7}$$

ここで $\text{count}(t)$ はテキストにおける t の出現回数を返す関数である．また，$p_{\text{base}}(t)$ は t の基本確率であり，文字レベルのバイグラム言語モデルから得られる．

$$p_{\text{base}}(t : c_1 ... c_M) = p_{\text{uni}}(c_1) \prod_{m=2}^{M} p_{\text{bi}}(c_m \mid c_{m-1}) \tag{8}$$

基本確率は語彙に含まれないトークンをサンプリングすることを可能にしており，$p_{\text{uni}}(c_m)$ と $p_{\text{bi}}(c_m \mid c_{m-1})$ もそれぞれ未知の文字ユニグラム，文字バイグラムを考慮して以下のように計算される．

$$\begin{aligned} p_{\text{uni}}(c_m) &= \frac{\text{count}(c_m) + \beta(\frac{1}{Y})}{Y + \beta} \\ Y &= \sum_{\hat{c}} \text{count}(\hat{c}) \end{aligned} \tag{9}$$

$$\begin{aligned} p_{\text{bi}}(c_m \mid c_{m-1}) &= \frac{\text{count}(c_m \mid c_{m-1}) + \gamma P}{\text{count}(c_{m-1}) + \gamma} \\ P &= p_{\text{uni}}(c_m) * p_{\text{uni}}(c_{m-1}) \end{aligned} \tag{10}$$

ここで Y はテキストに含まれる文字ユニグラムの総数を表し，$\text{count}(c_m \mid c_{m-1})$ は文字バイグラムの出現回数を返す関数である．また α，β 及び γ は語彙に含まれないトークンの確率を調整するハイパーパラメータである．この値が高いほど，未知語と既知語の確率の差が小さくなり，より多くの未知語をサンプリングできるようになる．

この言語モデルは MeCab などによる分割を用いて初期化する．また，教師なし単語分割と同様の手法 [6] を用いることで，辞書などを用いない初期化も可能である．

3.3 サンプリングと更新

学習時には言語モデル (7) による尤度に基づいて分割をサンプリングする．あらゆる分割候補を考慮した効率的なサンプリング手法として，Forward Filtering Backward Sampling[7] を用いる．これにより，トークンの最大長を固定することで高速に分割をサンプリングすることが可能である．

また言語モデルの更新はギブスサンプリングによる教師なし単語分割手法と同様に行う．ある文の分割をサンプリングする時，前回の分割によるパラメータを言語モデルから消去し，サンプリング後に新たなトークンの頻度をパラメータに追加する．更新後の言語モデルを用いてテストデータを分割することで，学習時の分割を考慮した文の分割が可能となる．

3.4 キャッシュを用いた埋め込み

本手法では可能な全ての分割を考慮するため語彙が固定されておらず，(1) のようなトークンの埋め込みを行うことができない．そこで，キャッシュを用いたトークンレベルの分散表現の獲得 [3, 1] を試みる．

キャッシュを用いた埋め込みでは，直近に出現したトークンを保存するキャッシュ$\boldsymbol{Q}$を保持し，キャッシュに含まれるトークンについては固有のベクトルをトークンレベルの分散表現として与える．具体的にあるトークン t は，$\boldsymbol{Q}$ に対応する分散表現の行列 $\boldsymbol{V}^{\text{cache}}$ を用いて以下のように得られる．

$$\boldsymbol{v}_t^{\text{token}} = \begin{cases} \boldsymbol{V}^{\text{cache}}\boldsymbol{k}_t & (t \in \boldsymbol{Q}) \\ \boldsymbol{v}_t^{\text{char}} & (\text{otherwise}) \end{cases} \tag{11}$$

t が $\boldsymbol{Q}$ に含まれない場合，キャッシュのうち最も古いトークンとこれに対応するベクトルを削除し，t をキャッシュに追加する．また，t に対応する $\boldsymbol{v}_t^{\text{token}}$ を $\boldsymbol{v}_t^{\text{char}}$ で初期化する．

得られた $\boldsymbol{v}_t^{\text{token}}$ は $\boldsymbol{v}_t^{\text{char}}$ とともに (4) を用いて，トークンの分散表現を計算する．キャッシュを用いた埋め込み $\boldsymbol{v}_t^{\text{token}}$ は，通常の Lookup 埋め込みと同様に学習データから最適化を行う．

4 実験

4.1 データセット

日本語，中国語および英語の感情分析データセットを用いて提案手法の評価を行った．フォーマルな文体のコーパスとして，NTCIR-6 意見分析パイロットタスク [8] を全ての言語で用いた．3 クラスの感情ラベルが付与された文のみを抽出し，32 トピックのうち 1-26 を学習データ，27-29 を開発データ，30-32 をテストデータとして使用した．ただし日本語は毎日新聞のデータのみを使用し，英語はコーパスのバランスを考慮し 1-28，29-30，31-32 をそれぞれ学習，開発，テストに用いた．

砕けた文体のコーパスとして，Twitter 感情分析データセットを日本語[1]と英語[2]について用いた．日本語 Twitter 感情分析データセットは 5 値，英語は 2 値の分類タスクで，どちらもバランスよく 21,000 文を抽出し使用する．さらに，中国語のデータセットとして ChnSentiCorp(HOTEL)[3]を利用した．

4.2 モデル設定

提案手法を評価するために，サブワード分割を比較対象とする．サブワード分割には SentencePiece[4][5] を使用し，サブワードサイズは英語 NTCIR データセットについては 6,000，その他のデータセットには 8,000 を用いる．また，サブワードのサンプリングを行う，行わない双方の設定を用いた．

異なる比較対象として，辞書による正解の分割をベースラインモデルとして用いた．日本語の辞書分割には MeCab[5]，中国語には Jieba[6]，英語の正解分割にはスペースによる単語境界を利用した．

提案手法による分割の最大単語長は言語によらず 8，ハイパーパラメータは $\alpha = \beta = \gamma = 1$ とした．キャッシュサイズはサブワードと条件を揃えるため，英語 NTCIR データセットについては 6,000，その他については 8,000 を設定した．提案手法の言語モデルは正解分割で初期化されたものと，500 エポックの教師なし分割により初期化されたものを用いた．

トークンベクトルは (4) を用いて，文字とトークン双方の情報から計算する．文字分散表現とトークンの分散表現はランダムに初期化し，学習データを用いて最適化する．

文字分散表現およびトークン分散表現のサイズはそれぞれ 128，512 とした．また，ベクトル結合を用いて文字とトークン双方から計算されるトークンベクトルのサイズも 512 である．文分散表現のサイズは 1,024 とし，トークンベクトルと文分散表現には 50% のドロップアウトを適用する．パラメータの最適化には Adam を使用し，開発データでの F1 値が最大となるモデルをテストデータで評価した．

[1] http://bigdata.naist.jp/~ysuzuki/data/twitter/
[2] https://www.kaggle.com/c/twitter-sentiment-analysis2
[3] http://tjzhifei.github.io/resource.html
[4] https://github.com/google/sentencepiece
[5] http://taku910.github.io/mecab/
[6] https://github.com/fxsjy/jieba

	Japanese		Chinese		English	
	NTCIR	TWITTER	NTCIR	HOTEL	NTCIR	TWITTER
gold	0.5554	0.65	0.5155	0.8528	0.5619	0.714
subword	0.5287	0.6625	0.5232	0.8645	0.5142	0.7265
subword+smp	0.5136	0.6625	**0.5342**	**0.8761**	0.5238	0.7315
proposed+sp	**0.5827**	0.665	0.5091	0.8662	**0.6095**	0.719
proposed+unsp	0.5307	**0.6775**	0.4954	0.8729	0.5523	**0.748**

表 1: 感情分析タスクにおける F1 値．太字は全体での最大，下線は正解分割を用いない手法での最大スコアを表す．

4.3 結果と考察

実験結果を表 1 に示した．辞書分割によるベースラインモデルを “gold”，サブワード分割を “subword”，サンプリング有りの設定を “subword+smp” と示している．また提案手法は正解分割を用いて言語モデルの初期化をしたものを “proposed+sp”，教師なし分割によって初期化したものを “proposed+unsp” と表記している．

正解分割を用いない手法（“subword”，“subword+smp” および “proposed+unsp”）における最大スコアは下線で示されている．結果より，提案手法が日本語と英語において最大となっており，中国語ではサブワード正則化（“subword+smp”）が最大となっている．提案手法の性能が中国語において低い理由として，中国語は ngram の種類が多いため，提案手法のキャッシュサイズが足りていないことが考えられる．

実験のうち，正解分割を用いた手法を含めた最大値は太字で示されている．多くのデータセットにおいて “proposed+sp” のスコアが “gold” を上回っており，正解分割が利用できる状況において提案手法がモデルの性能に良い影響を与えていることがわかる．また “proposed+sp” の性能は “gold” のスコアと関係があり，“gold” のスコアが高いデータセットにおいて提案手法を用いることで，大きな性能の向上が得られていることが確認された．

5 結論

ニューラルネットワークを用いた感情分析タスクにおいて，語彙を制限しない確率的な単語分割を行う手法について提案した．言語モデルを用いた学習データのサンプリングと，サンプリングに応じた言語モデルの更新を用いることで，日本語および英語の感情分析タスクの精度向上が確認された．今後は提案手法がスパム判定やトピック分類などの他の分類タスク，さらに NER や QA などの自然言語処理タスクにおいても有効であるかを検証して行きたい．

参考文献

[1] 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.

[2] Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 673–680. Association for Computational Linguistics, 2006.

[3] Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. *arXiv preprint arXiv:1612.04426*, 2016.

[4] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

[5] 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, 2018.

[6] 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.

[7] Steven L Scott. Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, Vol. 97, No. 457, pp. 337–351, 2002.

[8] 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. 2007.