大域的な類似度と部分文字列を用いた未知語分散表現の生成手法

五十川真生 † 梶原智之 ‡ 荒瀬由紀 †
† 大阪大学大学院情報科学研究科 ‡ 大阪大学データビリティフロンティア機構
{isogawa.mao, arase}@ist.osaka-u.ac.jp, kajiwara@ids.osaka-u.ac.jp

1 はじめに

対話システムや文書分類など、多くの自然言語処理タスクにおいて単語分散表現は基盤となる言語資源である。fastText [1] などの教師なし学習に基づく手法は、大規模コーパスからあらゆるタスクに適用可能な汎用的な単語分散表現を獲得することを目的とする。一方で、多くの応用タスクではモデル全体をラベル付きコーパス上で訓練する中でタスクに特化した単語分散表現を獲得することが多い。[*1]しかし、訓練データが語彙全体を網羅することは不可能であり、訓練済みモデルを実際に利用する際にはモデルの語彙に存在しない未知語を何らかの方法で処理する必要がある。このような未知語の問題は自然言語処理タスクが抱える重要な課題である。この問題を解決するため、本研究では所与の単語分散表現を模倣することによって、未知語の分散表現を動的に生成することを目的とする。

先行研究では、文字 [2, 3] や文字 N-gram [4] を入力とするニューラルネットワークを用いて、所与の単語分散表現を模倣している（図 1）。所与の単語分散表現を模倣することで、未知語であってもその文字や文字 N-gram をモデルに入力することで分散表現を生成できる。しかし、これらの手法によって得られる単語分散表現は品質が不十分である。例えば、単語分散表現の品質評価のためのベンチマークである単語間の意味的類似度推定タスク [5] において、手法 [2] は fastText の半分程度の性能に留まる。

本研究では、単語分散表現の模倣タスクにおいて 2 つの改善を行う。まず、「単語の意味は他の単語との関係によって定義される」と仮定し、単語分散表現を模倣するだけでなく単語分散表現間の関係も模倣する。先行研究はいずれも目標の 1 単語という局所的な情報のみを用いるが、我々の提案手法では目標の単語および他の単語集合という大域的な情報を用いて模倣モデルを訓練する。次に、我々は文字や文字 N-gram ではなく、Byte Pair Encoding (BPE) [6] に基づく部分文字列を入力として模倣モデルを訓練する。文字や大部分の文字 N-gram は意味的なまとまりを表現しないと考えられるが、BPE では単語を高頻度な部分文字列に分割するため接頭辞や接尾辞のような意味的なまとまりを抽出できる可能性がある。単語間の意味的類似度推定タスクにおける評価の結果、上記 2 点の工夫は模倣した単語分散表現の品質を既存手法と比較して有意に改善することを確認した。

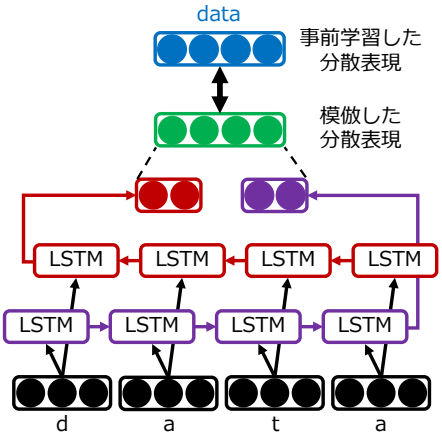

図 1: 単語分散表現の模倣タスク

2 関連研究

所与の単語分散表現を模倣する手法には、文字に基づく Reccurent Neural Network (RNN) [2] および Convolutional Neural Network (CNN) [3] や文字 N-gram に基づく Bag-of-Words [4] を用いた手法がある。本研究では、単語間の関係を模倣する大域的な訓練および BPE に基づく部分文字列の利用という 2 つの工夫によって、単語分散表現の模倣モデルを改善する。

機械翻訳などのテキスト生成タスクでは、単語の代わりに文字や BPE に基づく部分文字列 [6] を利用することが多い。低頻度な単語を文字や部分文字列にフォールバックすることで、未知語が出現しなくなる。

[*1] 上述の汎用的な単語分散表現を初期値とする場合も多い。

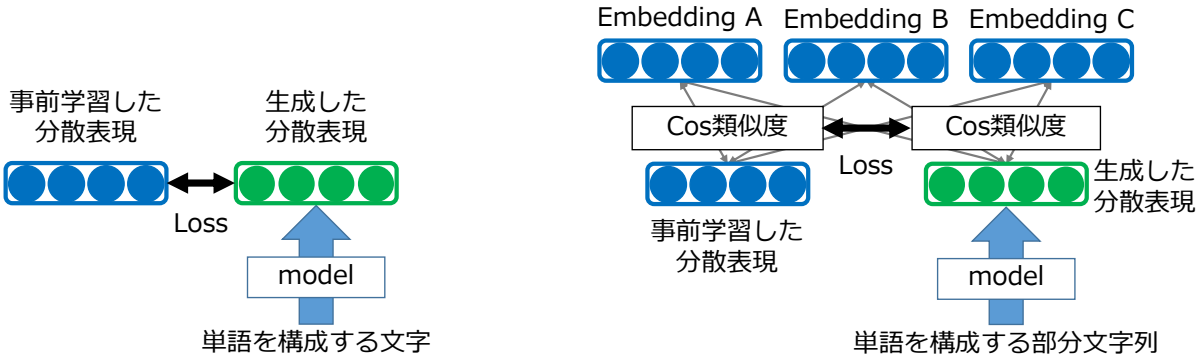

図 2: 提案手法と既存手法の対比

文字のみを用いると、語彙サイズを削減できる一方で文長が増加するという問題がある。そこで、語彙サイズと文長のバランスをとることができる部分文字列が広く用いられている。Zhao ら [4] は、単語分散表現の模倣タスクにおいて部分文字列を利用している。しかし、彼らは部分文字列として文字 N-gram を利用しているため、語彙サイズが増加してしまう。また、文字 N-gram に基づく部分文字列は意味の単位とは一致しない場合が多い。本研究では BPE に基づく高頻度な部分文字列を利用するため、接頭辞や接尾辞などの意味的な単位を捉えられる可能性がある。

3 提案手法

我々の提案手法の概要を図 2 に示す。本研究では、所与の単語分散表現を模倣するタスクにおいて、BPE に基づく部分文字列を入力として（3.1 節）、RNN によって模倣モデルを構築し（3.2 節）、単語分散表現間の関係を模倣する（3.3 節）。

3.1 BPE による部分文字列分割

まず、BPE [6] によってコーパスから部分文字列の分割規則を学習しておく。また、部分文字列の分散表現は、fastText [1] のアルゴリズムを用いて事前学習しておく。この事前学習によって、部分文字列の分散表現には文字のみを用いる場合よりも豊富な情報を付与できると期待できる。単語分散表現を模倣する際には、所与の単語を部分文字列に分割し、3.2 節の模倣モデルへ入力する。

3.2 RNN に基づく模倣モデル

我々の模倣モデルは先行研究 [2] と同じく、双方向 RNN と 2 層の MLP によって構成される。単語 w に関する長さ k の部分文字列の系列が与えられたとき、順方向および逆方向の LSTM における最終隠れ層を $\boldsymbol{h_f^k} \in \mathbb{R}^\lambda$ および $\boldsymbol{h_b^0} \in \mathbb{R}^\lambda$（$\lambda$ は隠れ層の次元数）として、模倣した n 次元の単語分散表現 $\boldsymbol{v_m} \in \mathbb{R}^n$ は以下のように表される。

$$\boldsymbol{v_m} = \boldsymbol{O_T} \tanh(\boldsymbol{T_h}[\boldsymbol{h_f^k}; \boldsymbol{h_b^0}] + \boldsymbol{b_h}) + \boldsymbol{b_T} \quad (1)$$

ただし、$\boldsymbol{O_T} \in \mathbb{R}^{n \times \delta}$、$\boldsymbol{T_h} \in \mathbb{R}^{\delta \times 2\lambda}$、$\boldsymbol{b_h} \in \mathbb{R}^\delta$、$\boldsymbol{b_T} \in \mathbb{R}^n$（$\delta$ は隠れ層の次元数）は MLP のパラメタである。

3.3 単語分散表現間の関係の模倣

先行研究 [2] と同じく、所与の単語分散表現 $\boldsymbol{v_t}$ と模倣した単語分散表現 $\boldsymbol{v_m}$ の平均二乗誤差を最小化することで、3.2 節の模倣モデルを訓練する。目的関数 L_{local} は、分散表現 $\boldsymbol{v}$ の i 番目の要素を v^i とすると以下のように表される。

$$L_{\text{local}} = \frac{1}{n} \sum_i^n (v_t^i - v_m^i)^2 \quad (2)$$

これに加えて本研究では、N 個の学習済み分散表現の集合 V_c 中の各分散表現と分散表現 $\boldsymbol{v_t}$ との余弦類似度、V_c 中の各分散表現と分散表現 $\boldsymbol{v_m}$ との余弦類似度を近づけることで、模倣モデルを最適化する。

$$L_{\text{global}} = \frac{1}{N} \sum_{\boldsymbol{v_c} \in V_c}^N (\cos(\boldsymbol{v_t}, \boldsymbol{v_c}) - \cos(\boldsymbol{v_m}, \boldsymbol{v_c}))^2 \quad (3)$$

ただし、分散表現の集合 V_c は計算量を抑えるため、全ての語彙を使用するのではなく、語彙全体から N 語

をサンプリングする。しかし、N 語を無作為に選択すると対象単語 w と無関係な単語が多く収集され、ノイズとなるおそれがある。そこで、半分は余弦類似度の高い順に集め、残り半分は無作為に集める。

最終的には、以下の目的関数 L を最小化する。

$$L = L_{\text{local}} + L_{\text{global}} \tag{4}$$

4 実験

本研究では、fastText [1] の学習済み単語分散表現を模倣し、単語間の意味的類似度推定タスク [5] において提案手法の有効性を検証した。

4.1 実験設定

NLTK[*2]を用いてトークナイズされた Wikipedia[*3] の本文を使って fastText[*4]を訓練した。そして、出現頻度が 5 回以上の単語を対象として、200 次元の単語分散表現を得た。このうち高頻度な 10 万語の中から、98,000 件の訓練用データと 1,000 件の開発用データを無作為に抽出した。

BPE の分割規則は、トークン数を 32,000 として、同じく Wikipedia の本文を用いて訓練した。また、単語間の関係を模倣するために、本研究では語彙から $N = 100$ 単語をサンプリングして訓練した。最適化関数には adam を使用した。その他の設定は表 1 に示す。

評価には、単語分散表現の品質評価のためのベンチマークである単語間の意味的類似度推定タスク[*5]を用いた。本研究では、Rubenstein-Goodenough dataset (RG)、Miller-Charles dataset (MC)、Word Similarity 353 dataset (WS)、MEN dataset (MEN) および Stanford Rare Word Similarity dataset (RW) の 5 つのデータセットを利用した。

分散表現の模倣において部分文字列を利用する効果を検証するため、模倣モデルとして、Character-RNN と 2 層の MLP を用いるモデル（MIMICK [2]）、Character-CNN と 2 層の Highway Network を用いるモデル（GWR [3]）および文字 N-gram の分散表現を平均したものを用いるモデル（BoS [4]）と、我々の提案手法である Subword-RNN を比較した。また、単語分散表現間の関係を模倣する効果を検証するため、各モデルの損失関数に式 (4) を適用したものと比較した。

[*2]http://www.nltk.org/
[*3]https://dumps.wikimedia.org/enwiki/20180801/
[*4]https://github.com/facebookresearch/fastText/
[*5]https://github.com/mfaruqui/eval-word-vectors/

表 1: 実験設定

δ	1200
λ	600
文字の分散表現の次元数	20
部分文字列の分散表現の次元数	200
バッチサイズ	50
BoS のエポック数	100
BoS 以外のエポック数	40

4.2 実験結果

表 2 に単語間の意味的類似度推定タスクにおけるスピアマンの順位相関係数を示す。ここで、fastText は模倣先の単語分散表現であるため、各手法の性能の上界を示している。部分文字列を用いることで、提案手法は先行研究における模倣モデルの性能を大きく上回った。また、MIMICK および GWR の各モデルに単語分散表現間の関係を考慮する L_{global} を加えることで、性能が向上することを確認した。提案手法ではこれら 2 つの効果により、RW を除く全てのデータセットにおいて最も高い性能を達成した。

5 考察

5.1 単語分散表現間の関係を模倣する効果

分析の結果、単語分散表現間の関係を模倣する式 (4) の損失関数を用いると、模倣モデルによって推定される単語間の意味的類似度と fastText によって推定される単語間の意味的類似度の誤差が減少することが明らかとなった。例えば MIMICK の損失関数を式 (4) に変更すると、fastText によって推定される単語間の意味的類似度との平均絶対誤差の micro 平均が 0.187 から 0.170 に減少した。これは、我々の期待通り、式 (4) の損失関数を用いることで所与の単語分散表現における単語間の関係を上手く模倣できるようになることを意味する。

一方、BoS に対しては式 (4) の目的関数が有効に機能しなかった。BoS の損失関数を式 (4) に変更すると、fastText によって推定される単語間の意味的類似度との平均絶対誤差の micro 平均が 0.101 から 0.107 に増加した。これは、BoS は分散表現を平均する単純なモデルであるため、限定的な表現力しか得られないためと考えられる。

表 2: 単語間の意味的類似度推定タスクにおけるスピアマンの順位相関係数

	RG	MC	WS	MEN	RW	micro 平均
Character-RNN (MIMICK)	0.428	0.394	0.397	0.390	0.335	0.371
Character-CNN (GWR)	0.413	0.392	0.409	0.409	0.342	0.384
Bag of N-gram (BoS)	0.734	0.663	0.550	0.644	0.218	0.481
Subword-RNN	0.744	0.782	0.661	0.694	**0.398**	0.583
Character-RNN (MIMICK) + Global	0.534	0.330	0.403	0.470	0.340	0.417
Character-CNN (GWR) + Global	0.500	0.524	0.421	0.460	0.355	0.419
Bag of N-gram (BoS) + Global	0.743	0.707	0.493	0.627	0.227	0.472
Subword-RNN + Global	**0.785**	**0.809**	**0.703**	**0.720**	0.391	**0.598**
fastText	0.821	0.808	0.716	0.754	0.481	0.651

5.2 部分文字列に関する事前学習の効果

MIMICK と Subword-RNN は、入力のみが異なる。つまり、模倣モデルの入力を文字の分散表現から事前学習した部分文字列の分散表現に変更すると、単語間の意味的類似度推定タスクにおけるスピアマンの相関係数が 0.371 から 0.583 まで改善される。しかし、部分文字列の分散表現を事前学習によって初期化しない場合、その性能は 0.030 まで低下する。これは、訓練データ中に出現する各部分文字列の出現頻度が少ないためである。BPE によって分割される部分文字列 32,000 個中、訓練データに出現しない部分文字列は 6,772 個、出現頻度が 5 回以下の部分文字列は 26,876 個あり、各部分文字列の平均出現頻度は 6.44 回である。しかし、文字の場合、最低頻度のものでも 1,254 回出現し、平均出現頻度は 27,434 回ある。そのため、部分文字列は文字よりも学習が難しいことがわかる。

6 おわりに

タスクに特化した単語分散表現を用いる際に発生する未知語問題を解決するために、本研究では所与の単語分散表現を模倣するモデルを改良した。我々のモデルは、BPE に基づく部分文字列を入力し、単語分散表現間の類似度を模倣する大域的な損失関数によって最適化する。これにより所与の単語分散表現だけでなく単語分散表現間の関係まで模倣でき、高品質な単語分散表現を動的に生成できる。

単語分散表現の品質評価のための代表的なベンチマークである単語間の意味的類似度推定タスクにおいて提案手法の有効性を検証した。今後は、分類や生成などの応用タスクおよびドメイン適応において模倣した単語分散表現を適用することで、それらのタスク全体の性能における効果について調査したい。

参考文献

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *TACL*, Vol. 5, pp. 135–146, 2017.

[2] Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. Mimicking Word Embeddings using Subword RNNs. In *Proc. of EMNLP*, pp. 102–112, 2017.

[3] Yeachan Kim, Kang-Min Kim, Ji-Min Lee, and SangKeun Lee. Learning to Generate Word Representations using Subword Information. In *Proc. of COLING*, pp. 2551–2561, 2018.

[4] Jinman Zhao, Sidharth Mudgal, and Yingyu Liang. Generalizing Word Embeddings using Bag of Subwords. In *Proc. of EMNLP*, pp. 601–606, 2018.

[5] Manaal Faruqui and Chris Dyer. Community Evaluation and Exchange of Word Vectors at wordvectors.org. In *Proc. of ACL*, pp. 19–24, 2014.

[6] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proc. of ACL*, pp. 1715–1725, 2016.