ニューラルネットワークによる教師なし単語分割

友利 涼† 森 信介††
†京都大学大学院 情報学研究科
††京都大学 学術情報メディアセンター
tomori.suzushi.72e@st.kyoto-u.ac.jp, forest@i.kyoto-u.ac.jp

1 はじめに

日本語や中国語のように単語の境界が明示的に与えられない言語では、単語分割は構文解析などの後段の処理に必要なタスクであり、それらの言語の自然言語処理において不可欠な技術になっている。一般的な単語分割では、大量のアノテーション付きテキストを用いた教師あり学習手法がよく用いられる。しかし、単語分割が付与されたコーパスが存在するドメインや言語は限られており、アノテーション付きテキストを必要としない単語分割手法が求められている。

教師なし単語分割の研究には Nested Pitman-Yor Language Model (NPYLM) に基づく単語分割 [3] が提案されている。これは観測された文字列を単語 n-gram 言語モデルが最適 (予測力最大) となる単語列へ分割するノンパラメトリックベイズ手法である。しかし、近年、リカレントニューラルネットワーク (RNN) による言語モデルが提案されており、n-gram 言語モデルよりも性能が良いことが知られている。そこで、RNN 言語モデルを用いた教師なし単語分割を提案する。

2 関連研究

NPYLM[3] は分かち書きされていない文を単語 n-gram 言語モデルが最適 (予測力最大) となる単語列へ分割する手法である。これは階層的 Pitman-Yor 言語モデル (HPYLM)[5] を拡張することで文字列から言語モデルを構築し、言語モデルを最適化するような単語列を得ることができる。

HPYLM は Pitman-Yor 過程に基づくノンパラメトリックベイズの n-gram モデルであり、以下の式のように $(n-1)$-gram 分布が n-gram 分布を生成すると仮定している。

$$G_n \sim \mathrm{PY}(G_{n-1}, d_n, \theta_n)$$

ここで、d_n はディスカウント係数、θ_n は G_n が平均的に基底測度 G_{n-1} に似ているかを制御するパラメータである。これによりユニグラム分布からバイグラム分布が、バイグラム分布からトライグラム分布が生成される。実際に文脈 h が与えられたときの n-gram 確率は、文脈 h が与えられたときの w の出現回数 $p(w|h)$ と単語 w が 1 つ短い文脈 h' から生成されたと推定された回数 t_{hw} を用いて以下の式から再帰的に計算される。

$$p(w|h) = \frac{c(w|h) - d \cdot t_{hw}}{\theta + c(h)} + \frac{\theta + d \cdot t_{h\cdot}}{\theta + c(h)} p(w|h')$$

ここで、$t_{h\cdot} = \Sigma_w t_{hw}$, $c(h) = c(w|h)$ とした。

NPYLM では部分文字列の単語らしさを推定するために、HPYLM を ∞-gram モデルに拡張した VPYLM [2] を用いて文字モデルを構築し、単語モデルの HPYLM のゼログラム基底測度に文字モデルの VPYLM をネストする。学習の際には MCMC 法を使い、前向き後向きアルゴリズムにより確率的に単語分割を行う。最大単語長を L としたとき、部分文字列 $c_1 c_2 \cdots c_t = c_1^t$ の最後の k 文字を単語として生成する前向き確率 $\alpha[t][k]$ は以下の式で表され、文末まで再帰的に計算していく。

$$\alpha[t][k] = \sum_{j=1}^{L} p(c_{t-k+1}^{t} | c_{t-k-j+1}^{t-k}) \alpha[t-k][j] \quad (1)$$

単語分割位置のサンプリングは、文末から後ろ向きに求まる。文末の特別な記号を EOS とし入力文字列の長さを N とすると、$p(\mathrm{EOS}|c_{N-k}^{N})\alpha[N][k]$ に比例する確率で k をサンプリングし、文字列の最後の単語が得られる。その前の単語も今決めた単語に前接するようにサンプリングでき、これを文字列の先頭まで繰り返すことで入力文字列から単語列をサンプリングできる。サンプリングした単語列を通常の HPYLM と同様に MCMC の中で削除と再追加を繰り返すと言語モデルを最適化するようなパラメータが得られる。

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

Pitman-Yor Hidden Semi-Markov Model (PYHSMM) による教師なし形態素解析も提案されている [6]。これは NPYLM を拡張したモデルで、単語の潜在クラスを品詞とみなし、品詞 n-gram と品詞ごとの単語生起確率を計算することで単語とその品詞をサンプリングする。

3 提案手法

本手法は、観測文字列 $c_{1:N} = c_1, c_2, ..., c_n, ..., c_N$ が与えられたとき、RNN 言語モデルが最適となる単語列 $w_{1:T} = w_1, w_2, ..., w_t, ..., w_T$ を推定する。提案手法の概要を図 1 に示す。

3.1 強化学習によるパラメータ更新

本手法は強化学習に基づいている。強化学習はある環境に対して探索とパラメータ更新を繰り返して環境に適応する学習方式である。$c_{1:n}$, $w_{1:t}$ まで解析済みの場合、現在の状態 $\overrightarrow{\boldsymbol{h}}_t^w$ から行動 a_t を選択し、報酬 r_t を受け取り次の状態 $\overrightarrow{\boldsymbol{h}}_{t+1}^w$ に遷移する。ここでの状態 $\overrightarrow{\boldsymbol{h}}_t^w$ は RNN 言語モデルの隠れ状態を指し、行動 a_t は次の単語 w_{t+1} を $c_{n+1}, c_{(n+1):(n+2)}, ..., c_{(n+1):(n+L)}$ から選択できる。ここで L は単語となりうる文字列の最大の長さである。言語モデルのスコアを式 (2) とすると、次の単語 w_{t+1} を選択したときの報酬 r_t は、これまでの $w_{1:t}$ を用いて式 (3) と計算する。ここで $\mathrm{MC}^\theta(w_{1:t}, c_{1:N}; S)$ は、パラメータ θ に従うモンテカルロ探索により文末までの単語列を S 回サンプリングすることであり、これにより言語モデルのスコアの期待値が計算できる。

$$\mathrm{LM}(w_{1:T}) = \frac{1}{T}\sum_{t=1}^{T} p(w_t|w_{1:(t-1)}) \tag{2}$$

$$r_t = \frac{1}{S}\sum_{s=1}^{S} \mathrm{LM}(w_{1:T}^s), w_{1:T}^s \in \mathrm{MC}^\theta(w_{1:t}, c_{1:N}; S) \tag{3}$$

学習では探索とパラメータ更新を繰り返しながら報酬の和を最大化するような方策を求める、つまり言語モデルが最適となる単語列を求めるパラメータ θ を学習する。実際の学習には、行動 a_t を選択した場合の言語モデルの期待値 r_t と方策勾配法を用いることで、言語モデルを最適化するような行動を学習することができ、単語の生成確率と報酬 r_t の積から勾配を計算できる。

3.2 単語の生成確率

単語の生成確率 $p(w_{t+1}|w_{1:t})$ の計算において、一般的な RNN 言語モデルでは語彙サイズが制限されてしまうが、教師なし単語分割ではすべての部分文字列が単語となりうるため語彙サイズはあらかじめ設定できない。そこで我々はオープンボキャブラリーによる RNN 言語モデル [1] を拡張する。この手法は直前に出現した v 個の単語の分散表現をキャッシュとして保持し、式 (4) のように適当な値 g を用いて文字レベルの言語モデルでスムージングを行う。

$$p(w_{t+1}|\overrightarrow{\boldsymbol{h}}_t^w) = (1-g)p_w(w_{t+1}|\overrightarrow{\boldsymbol{h}}_t^w) + gp_c(w_{t+1}) \tag{4}$$

我々は常に v 個の単語を保持するのではなく、NPYLM のようにサンプリングした単語の削除・追加を行うことで動的に変化する語彙集合 V とそれらの単語の分散表現の行列 $\boldsymbol{V}^\theta$ を得る。語彙集合 V に w_{t+1} が存在するならば、$p_w(w_{t+1}|\overrightarrow{\boldsymbol{h}}_t^w)$ の計算は、以下のように単語の分散表現の行列 $\boldsymbol{V}^\theta$ とクエリベクトル $\boldsymbol{q}_t$ の積から計算できる。

$$p_w(w_t|\overrightarrow{\boldsymbol{h}}_t^w) = \mathrm{softmax}_{w_t}(\boldsymbol{V}^\theta \boldsymbol{q}_t)$$
$$\boldsymbol{q}_t = \tanh(\boldsymbol{W}_q \overrightarrow{\boldsymbol{h}}_t^w + \boldsymbol{b}_q)$$

スムージング係数 g は、ハイパーパラメータ g' を $\boldsymbol{V}^\theta$ とクエリベクトル $\boldsymbol{q}_t$ の積のベクトルに連結し、softmax を計算したものを用いる。

$$g = \mathrm{softmax}_{g'}(\boldsymbol{V}^\theta \boldsymbol{q}_t : g')$$

ハイパーパラメータ g' の値が $\boldsymbol{V}^\theta$ とクエリベクトル $\boldsymbol{q}_t$ の積のベクトルの各要素に対し、相対的に高い値ならば文字レベルの言語モデルでスムージングする割合が高まる。つまり、次の単語が V のいずれでもないと予測した場合に g の値が大きくなる。

w_{t+1} を構成する文字列を $c_{n+1:n+k}$ とすると、$p_c(w_{t+1})$ は以下のように計算する。

$$p_c(w_{t+1}) = \frac{1}{2}\prod_{i=0}^{k-1} p(c_{n+1+i}|\overrightarrow{\boldsymbol{h}}_i^c) + \frac{1}{2}\prod_{i=0}^{k-1} p(c_{n+k-i}|\overleftarrow{\boldsymbol{h}}_i^c)$$

ここで $\overrightarrow{\boldsymbol{h}}^c$ は順方向の文字レベル言語モデルの隠れ状態であり、$\overleftarrow{\boldsymbol{h}}^c$ は逆方向の文字レベル言語モデルの隠れ状態である。

3.3 ギブスサンプリングによる学習

本手法は NPYLM と同様にギブスサンプリングを行うことで学習を行う。文字列 $c_{1:N}$ があたえられた

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

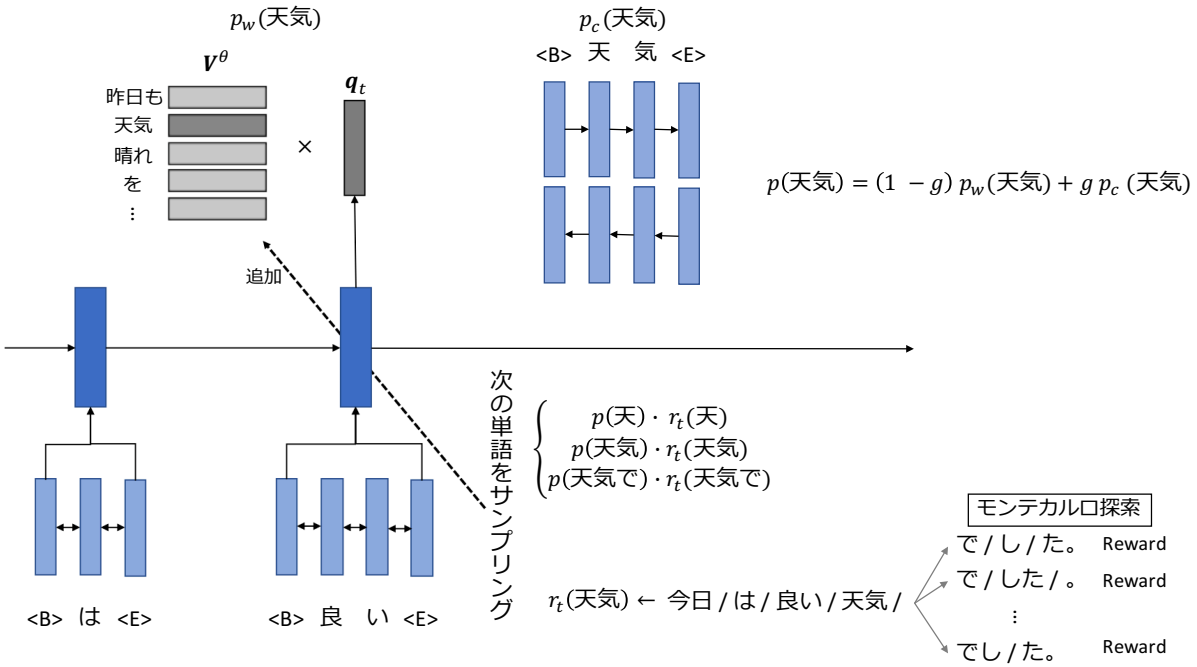

図 1: 提案手法の概要

とき、文頭の特別な記号 BOS の分散表現を入力し、$p(c_{1:k'}|\overrightarrow{\boldsymbol{h}}_0^w) \cdot r_0(c_{1:k'})$ に比例する確率で最初の単語 $c_{1:k'}$ をサンプリングする。その次の単語も同様に $c_{1:k'}$ の分散表現を入力し、サンプリングできる。これを文末まで繰り返す。

サンプリングした単語列は NPYLM と同様にパラメータとして削除・追加を行う。本手法では、サンプリングした単語列に対し、方策勾配法に基づき単語レベルの RNN の学習を行い、文字レベルの RNN では通常の RNN 言語モデルと同様に次の文字を予測するよう学習を行う。

V と $\boldsymbol{V}^\theta$ に単語の追加を行う場合、その単語が V に存在しないのであれば V と $\boldsymbol{V}^\theta$ に単語とその分散表現をそれぞれ追加する。その単語が語彙集合 V に存在するのではあればその単語の出現カウントを 1 増やし、その単語の分散表現を更新する。ここで単語の分散表現には以下のように、文字レベルの順方向と逆方向の最後の隠れ状態を連結して用いる。

$$\mathrm{Embed}(c_{n+1:n+k}) = \tanh(\overrightarrow{\boldsymbol{h}}^c_{k-1} : \overleftarrow{\boldsymbol{h}}^c_{k-1})$$

単語の削除を行う場合、その単語の出現カウントを 1 減らし、もし出現カウントが 0 になれば V と $\boldsymbol{V}^\theta$ から単語とその分散表現をそれぞれ削除する。

図 2 に学習アルゴリズムを示す。

for $epoch = 1$ to E **do**
  for $c_{1:N}, w_{1:T}$ = randperm(corpus) **do**
    if $epoch > 1$ **then**
      Remove parameters of $w_{1:T}$ from $V, \boldsymbol{V}^\theta$
    end if
    Sample $w_{1:T}, r_{1:T}$ according to $(c_{1:N}, \theta)$
    update RNN according to $w_{1:T}, r_{1:T}$
    Add parameters of $w_{1:T}$ to $V, \boldsymbol{V}^\theta$
  end for
end for

図 2: 教師なし単語分割の学習アルゴリズム

表 1: 将棋解説コーパスの諸元

	文数	文字数	文字種類数
学習 (分割なし)	2,041	46,065	1,167

4 実験

提案手法の有効性を確認するために、プログラムを実装[1]し実験を行った。表 1 に実験で用いたコーパスの諸元を示す。コーパスには将棋解説コーパス [4] を用いた。このコーパスの単語は UniDic の短単位の基準の活用語尾を分割した超短単位に基づいている。学習に用いた文の 491 文をテストの際に解析した。実験

[1] https://github.com/tomoris/unsupervised_nn_ws

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.

表 2: 単語分割結果

	再現率	適合率	F 値
NPYLM	26.09	41.24	31.96
PYHSMM	34.20	56.05	42.48
提案手法	60.72	56.10	58.32

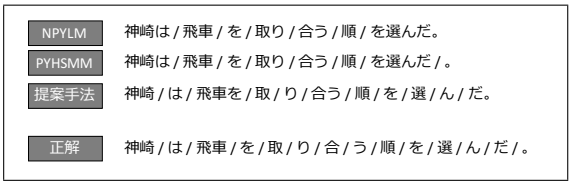

図 3: 提案手法の概要

では比較手法として NPYLM と PYHSMM を用いた。NPYLM は、PYHSMM のプログラムで品詞数 $Z = 1$ とし、PYHSMM では品詞数 $Z = 15$ とした。

提案手法のハイパーパラメータとして、文字の分散表現の次元数、文字レベルの順方向 RNN と逆方向 RNN の隠れ状態の次元数をそれぞれ 100 とし、単語レベルの順方向 RNN の次元数を 200 とした。RNN のネットワークモデルとして GRU を用いた。スムージング係数を決める $g' = 0$、最大単語長 $L = 8$ とした。また、提案手法は計算コストが大きく収束が遅いため、PYHSMM ですでに学習・分割された文字列を初期状態として用いた。また、報酬を求めるモンテカルロ探索も計算コストが大きいため、学習の際には $S = 1$ とした。テストの際には $S = 5$ として、$p(c_{n+1:n+k}|\overrightarrow{\boldsymbol{h}}_t^w) \cdot r_t(c_{n+1:n+k})$ が最も大きい文字列 $\mathrm{c}_{n+1:n+k}$を単語として分割した。

表 2 に実験結果を示す。提案手法の単語分割精度が比較手法よりも高いことがわかる。分割結果の一部を図 3 に示す。提案手法では、文献 [3, 6] のように単語長の補正を行わなかったため、高頻度に出現する文字列を除いて、1 文字単位で分割することが多かった。そのため、超短単位の基準に適応し、比較手法よりも精度が向上したと考えている。参考としてすべての文字間に単語境界があるとして分割した場合の再現率、適合率、F 値はそれぞれ 72.13、54.54、62.11 である。

5 おわりに

本稿では、ニューラルネットワークに基づく教師なし単語分割を提案した。本手法は、語彙サイズを制限しない RNN 言語モデルに基づいており、適当な値と文字レベルの RNN 言語モデルを用いてスムージングを行う。パラメータの更新には、ギブスサンプリングと強化学習を用いて言語モデルが最適となるように学習を行う。

比較手法よりも提案手法が単語分割を精度よく行えた。今後の課題として、半教師あり学習の際の精度変化を調査したい。ニューラルネットワークを用いることにより実世界情報を参照しやすくなったため、テキストに付随する実世界情報を参照した場合の分析も行いたい。また、本手法は計算コストが大きいため、高速化についても検討したい。今回は教師なしの単語分割なので、開発セットを用いてハイパーパラメータ等の調節は行わなかった。そのため、比較手法が超短単位の基準から大きく外れてしまった可能性がある。今後はハイパーパラメータを変化させたときの分析や異なるコーパスでの実験、単語長の補正なども行いたい。

謝辞

本研究を進めるにあたり、プログラムの一部を提供して頂いたデンソーアイティーラボラトリ及び同社の内海 慶氏に謝意を表する。

参考文献

[1] Kazuya Kawakami, Chris Dyer, and Phil Blunsom. Learning to create and reuse words in open-vocabulary neural language modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1492–1502, 2017.

[2] Daichi Mochihashi and Eiichiro Sumita. The infinite markov model. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pp. 1017–1024, 2007.

[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 Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 100–108, 2009.

[4] Shinsuke Mori, John Richardson, Atsushi Ushiku, Tetsuro Sasada, Hirotaka Kameko, and Yoshimasa Tsuruoka. A japanese chess commentary corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluatio*, pp. 1415–1420, 2016.

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

[6] Kei Uchiumi, Hiroshi Tsukahara, and Daichi Mochihashi. Inducing word and part-of-speech with pitman-yor hidden semi-markov models. 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)*, pp. 1774–1782, 2015.

Copyright(C) 2018 The Association for Natural Language Processing.
All Rights Reserved.