文字レベル言語モデルの転移学習による日本語単語分割

李 桐　　鶴岡 慶雅
東京大学 大学院情報理工学系研究科
{litong,tsuruoka}@logos.t.u-tokyo.ac.jp

1 はじめに

日本語や中国語のような分かち書きが自明ではない言語においては、入力した文を単語に分割する処理は欠かせない前処理の一つである。単語分割段階での誤りは後段のタスクに大きな影響を与えうるため、単語分割には高い精度が求められている。教師データを用いた系列ラベリングによる手法は単語分割のタスクの主流であるが [4] 、教師データは人手により構築しなければならないため、多くの場合はそのサイズが限られており、教師データに存在しない未知語に対応しきれない問題がある。

転移学習によって大規模なデータセットで事前学習を行ったモデルを他のタスクに転用する手法は、画像処理分野でよく使われており、自然言語処理においても有効であることが報告されている。転移学習を行う際には、主に２つのアプローチがある：

- **素性ベース**：事前学習したモデルのパラメータを固定し、素性抽出器として使い、抽出した素性を入力として用いて、タスクモデルをゼロから学習する手法。
- **ファインチューニング**：事前学習を行ったあと、そのパラメータを初期値としてし、他の教師ありデータで再度学習させ微調整する手法。

自然言語処理に分野おける転移学習では、素性ベースのアプローチが主流である。しかし最近、BERT [3] を代表する、膨大な数のパラメータでから成る言語モデルを大量のデータで事前学習し、ファインチューニングによって他のタスクに転用する研究がいくつか発表され、素性ベースの手法より高い性能を実現した。

このような背景を踏まえ、本研究では、日本語版 Wikipedia のデータで文字レベル言語モデルの事前学習を行い、得られたモデルを日本語単語分割のタスクに転移学習する手法を提案する。

2 関連研究

2.1 日本語単語分割

日本語単語分割のタスクは系列ラベリング問題として扱われ、従来の手法には

- **ラティスベース**：学習データまたは外部資源の語彙により作成した辞書を基づいて、単語ラティスを構築し、ラティス中のパスを探索する手法 [5]
- **文字ベース**：教師データを使って、独立に文字毎に単語境界のラベルを推定できるように学習する手法 [8]

の二種類がある。近年では、深層学習に関する研究が盛んでおり、ニューラルネットワークを日本語単語分割に取り込む手法もいくつか提案されている。Morita ら [7] は単語列の尤もらしさを評価するリカレントニューラルネットワーク言語モデルを単語ラティスに基づくベースモデルに加え、両方のスコアの重み付き和で単語列のスコアを計算する手法を提案し、F 値で 0.2 〜0.6 の向上を実現した。一方、Kitagawa ら [4] はリカレントニューラルネットワークの一種である Long Short Term Memory (LSTM) を用いたニューラル単語分割の手法を提案し、同じく外部知識を利用しない設定で、現代日本語書き言葉均衡コーパス (クラス A) において、F 値 98.42 と KyTea [8] を上回る性能を達成した。

2.2 言語モデルの転移学習

Word2vec [6] や GloVe [9] などで事前学習した埋め込み行列を使い、単語を分散表現に転換する手法は、ニューラルネットワークによる自然言語処理で広く使われており、自然言語処理における転移学習の主流である。近年、計算パワーの増加を背景に、言語モデルの目的関数で事前学習したモデルを他のタスクへ転移

学習を行う研究がいくつか報告されている。素性ベースのアプローチとして、Peters ら [10] は事前学習した LSTM ベースの双方向言語モデルを用いて、文脈情報を考慮した分散表現を獲得する手法 ELMo を提案し、従来手法で使用されている Word2vec や GloVe に基づく埋め込み層を置き換えるだけで、極性分類や固有表現抽出などのタスクで最高精度を達成した。Jacob ら [3] は、隣接文予測やマスク単語予測の 2 つの目的関数を使い、膨大な数のパラメータを持つ Transformer [11] ベースの言語モデルを数億文規模なデータセットで事前学習させたあと、ファインチューニングによって転移学習を行う手法を提案した。異なる 11 個の自然言語処理タスクに適用した結果、全てのタスクにおいて ELMo を超える精度を達成し、さらに SQuAD と呼ばれる機械読解のタスクにおいて F 値 93.2 を実現し、人間の精度を 2.0 ポイント上回った。

3 提案手法

単語分割とは、各文字の次に単語の境界があるかどうかを推定する問題であり、文字ごとに 0, 1 のラベルを予測する系列ラベリング問題とみなすことができる [8]。図 1 はその一例を示す。本研究では、この設定を基に、日本語 Wikipedia のデータで事前学習した文字レベル言語モデルを使い、素性ベースやファインチューニングの 2 つのアプローチにより、日本語単語分割タスクへの転移学習を行った。

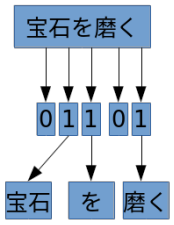

図 1: 文字ごと単語の境界の推定による単語分割

事前学習の部分は LSTM を用いて、文字レベルの双方向言語モデルの目的関数で最適化を行う。文字レベルの言語モデルは文 (文字の並び) に関する確率分布を評価するものである。ある文字の並びが自然であればその文の確率は高くなり、不自然であれば低くなる。文字の条件付き確率を計算するときに、前の文字の情報を利用する言語モデルは前向き言語モデル、後ろの文字の情報を利用する言語モデルは後ろ向き言語モデルと呼ばれる。

- 前向き言語モデル

$$p(w_1, w_2, \ldots, w_N) = \prod_{k=1}^{N} p(w_k | w_1, \ldots, w_{k-1})$$

- 後ろ向き言語モデル

$$p(w_1, w_2, \ldots, w_N) = \prod_{k=1}^{N} p(w_k | w_{k+1}, \ldots, w_N)$$

素性抽出による転移学習の仕組みは、図 2 に示すように、文を事前学習した言語モデルに入力し、同じ文字が前向きもしくは後ろ向き言語モデルにおけるの隠れ層状態を連結し、それを素性として利用する。

一方、ファインチューニングによって転移学習する際は、言語モデルの事前学習で使われていたソフトマックス層を二値分類問題用の層に置き換え、LSTM 層と新しく追加したソフトマックス層に別々の学習率を適用し、単語分割の学習データで再学習を行う。

4 実験

4.1 データ

本研究では、文字レベル言語モデルや単語分割用モデルの学習データとして、それぞれ日本語版 Wikipedia のダンプデータ (以下、wiki-dump) [1] と京都大学ウェブ文書リードコーパス (以下、KWDLC) [2]を利用する。wiki-dump では、wikiextractor [3]を使って文の抽出を行い、得られたデータの文数は 9,112,239 となった。KWDLC の方では、ランダムにデータセットを訓練データ、検証データ、テストデータに分割し、文数はそれぞれ 10,742、2,000、2,000 となった。

4.2 実験設定

本研究では、上述した文字レベル言語モデルの転移学習による実験設定のほか、同じ Wikipedia のデータで fastText [2] により学習した文字の埋め込みベクトルを素性として用いた単語分割モデルをベースラインとする。予備実験として、100、300、1024 次元の

[1] https://dumps.wikimedia.org/jawiki/20181120/
[2] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KWDLC
[3] https://github.com/attardi/wikiextractor

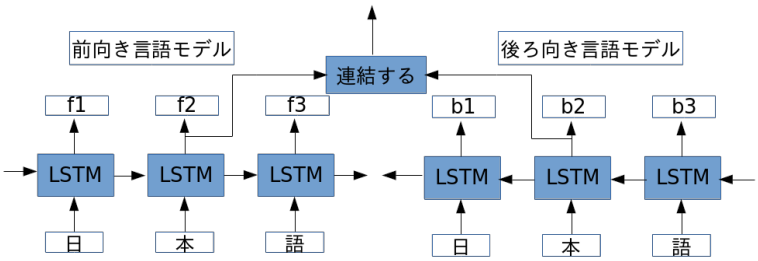

図 2: 文字レベルの双方向言語モデルによる素性の抽出

実験設定	文字埋め込み層次元数	LSTM 隠れ層次元数
一層 LSTM + fastText-1	100	256
一層 LSTM + 言語モデル素性	2048	256
一層 LSTM + fastText + 言語モデル素性	2148	256
二層 LSTM + fastText	100	256
ファインチューニング (一層 LSTM)	1024	1024
ファインチューニング (二層 LSTM)	1024	1024

表 1: 各実験設定のハイパーパラメータ

fastText 埋め込みベクトルをそれぞれ学習し、性能評価を行ったがところ、大きな差が観測されなかったため、実験結果では 100 次元の埋め込みベクトルによるもののみを示す。

文字レベル言語モデルの事前学習では、一層と二層の双方向 LSTM で構成したモデル 2 つを別々に学習した。訓練データのサイズはかなり大きく、メモリに載せるため 37 のスプリットに分割し、SGD で 10 エポックの学習を行う。ハイパーパラメータの設定は Akbik ら [1] の提案手法に参考する。学習率の初期値は 20 に設定し、学習損失が連続 25 スプリットで下がらなかった場合に学習率を四分の一に減衰する。言語モデルをファインチューニングする際は、Adam を使って最適化する。事前学習した LSTM の部分は学習率を 0.005 で、新しい追加したソフトマックス層は学習率 0.01 で再学習を行う。

素性ベースの単語分割のタスクモデルには、同じ双方向 LSTM モデルを利用する。予備実験で LSTM の隠れ層の次元数を 256, 512, 1024 に設定しそれぞれ評価を行ったところ、次元数の増大による性能の向上が観測されなかったため、本実験では全て 256 に設定する。最適化は Adam により行い、学習時に早期打ち切りと学習率減衰の手法を使う。具体的には、検証データによって算出した損失が連続 3 エポックで下がらなかった場合に学習率を半減し、学習率が 0.0001 以下になった時点で学習を打ち切る。

具体的なハイパーパラメータの設定を表 1 に示す。

4.3 結果

各設定で訓練データでモデルを学習し、検証データとテストデータで F 値により評価した結果と、その F 値に到達するまで学習したエポック数を表 2 に示す。

4.4 考察

素性ベースのアプローチによる言語モデルの転移学習では、ベースラインの埋め込みベクトルによるものに比較して、性能の向上を示せなかった。さらに、言語モデルで抽出した素性を埋め込みベクトルと連結して使った場合に、性能の低下が観測された。ここはさらなる実験で原因を究明する必要がある。ファインチューニングによる転移学習の実験においては、一層の LSTM で構成された言語モデルをベースにしたものはベースラインより下回る性能である一方、二層 LSTM の文字レベル言語モデルは、7 エポックの再学習ですべての設定において最も良い性能を実現した。これらの結果から、日本語単語分割のタスクに有用な知識は、文字レベル言語モデルの事前学習において獲得されたことがわかる。

実験設定	F 値 (検証データ)	F 値 (テストデータ)	学習エポック数
一層 LSTM + fastText	97.04	96.70	114
一層 LSTM + 言語モデル素性	97.07	96.66	148
一層 LSTM + fastText + 言語モデル素性	96.86	96.49	139
二層 LSTM + fastText	97.06	97.10	114
ファインチューニング (一層 LSTM)	96.03	95.53	4
ファインチューニング (二層 LSTM)	**97.37**	**97.13**	7

表 2: 各実験設定の F 値

5 おわりに

本研究では、Wikipedia のデータで事前学習を行った日本語文字レベル言語モデルを使い、素性抽出とファインチューングの 2 つのアプローチによって、単語分割タスクへの転移学習を試みた。今後の研究としては、解析結果での誤りや素性を連結して使うときに観測された性能低下の原因を分析することが挙げられる。また、今回は計算資源の都合により、事前学習の部分では LSTM ベースの双方向言語モデルしか試せなかったため、BERT [3] で高い精度を示した Transformer ベースのモデルやマスク単語予測の事前学習タスクで提案手法をさらに検証したい。

6 謝辞

本研究の一部は、独立行政法人情報通信研究機構（NICT）の委託研究「多言語音声翻訳高度化のためのディープラーニング技術の研究開発」の助成を受けて実施された。

参考文献

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018*.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *TACL*, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2018.

[4] Yoshiaki Kitagawa and Mamoru Komachi. Long short-term memory for japanese word segmentation. *arXiv:1709.08011*, 2017.

[5] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *EMNLP 2004*.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS 2013*.

[7] Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. Morphological analysis for unsegmented languages using recurrent neural network language model. In *EMNLP 2015*.

[8] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *HLT 11*.

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP 2014*.

[10] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL 2018*.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS 2017*.