Bi-LSTM CRF モデルを用いた平仮名文の形態素解析

井筒 順
茨城大学工学部情報工学科
17t4013g@vc.ibaraki.ac.jp

古宮 嘉那子
茨城大学 理工学研究科
kanako.komiya.nlp@vc.ibaraki.ac.jp

1 はじめに

自然言語処理の中で，形態素解析は根幹技術となっている。現代においては MeCab[1] や Chasen[2] 等の形態素解析システムが存在し，形態素解析の中核を担っている。しかし，上記システムは漢字仮名交じり文を対象にしているため，平仮名で構成された文章を形態素解析することは難しい。

本稿では Bi-LSTM CRF モデルを用いて平仮名文の形態素解析モデルを作成した。実験では，訓練事例のジャンルの影響を見るために，Wikipedia と Yahoo!知恵袋のデータの二種類の訓練事例を利用した。また，両データを利用してファインチューニングを行い，様々なジャンルのテキストにおける影響を調べた。

また，形態素解析は，文章を単語ごとに分割することに加え，分割した単語に対して品詞等の情報を付与する必要があるが，全て平仮名文であることから，漢字仮名交じり文と比べ単語ごとに分割する際の情報が少なくなり，単語分割の精度が低下してしまうことが予想される。そこで本稿では，漢字仮名交じり文によるモデルを用いたファインチューニングを行った。

本論文の貢献は以下の 4 つである。

- Bi-LSTM CRF モデルを用いて平仮名文の単語分割モデルを生成したこと
- 様々なジャンルのテキストを対象に，形態素解析の訓練事例の影響を検証したこと
- Wikipedia と Yahoo!知恵袋のデータを利用したファインチューニングの有効性を示したこと
- 漢字仮名交じり文によるモデルを用いたファインチューニングの有効性を示したこと

本論文ではこれらの実験の結果を報告する。

1） https://taku910.github.io/mecab/
2） https://chasen-legacy.osdn.jp

2 関連研究

工藤ら [1] は，平仮名交じり文が生成される過程を生成モデルでモデル化した。そして，そのパラメータを大規模 Web コーパス及び EM アルゴリズムで推定することにより，平仮名交じり文の解析精度を向上させる手法を提案している。

大崎ら [2] は，文章中の特徴的な表現を新たな名称として扱うことによって，コーパスの構築を行っている。また，藤田ら [3] は既存の辞書やラベルありデータを対象分野の特徴に合わせて自動変換し，それを使用することで形態素解析モデルを構築する教師なし分野適応手法を提案している。林ら [4] は，平仮名語の単語を辞書に追加することにより，形態素解析の精度が向上することを報告している。

また，井筒ら [5] は MeCab の ipadic 辞書を平仮名に変換し，平仮名のみで構成されたコーパスをを用いることで平仮名のみの文章での形態素解析を行っている。

Ma ら [6] は Bi-LSTM モデルを使用した中国語の単語分割モデルの作成を行なっている。そして単語分割の精度が，Bi-LSTM モデルよりも複雑なニューラルネットワークアーキテクチャに基づくモデルと比較して，一般的なデータセットでより良い精度を達成することを報告している。

また，Thattinaphanich ら [7] は Bi-LSMT CRF モデルを構築しタイ語において固有表現抽出を行っている。タイ語では，言語リソースが少ないことや，単語・フレーズ・文の境界指標がないなどの言語的な問題が存在するが，単語表現と Bi-LSTM を用意し CRF と組み合わせ，テキストのシーケンスを学習しその知識を利用することでこの問題を克服している。

3 提案手法

本稿では，Bi-LSTM CRF モデルを用いて平仮名文の単語分割モデルを生成する。実験では，

Wikipedia モデル，BCCWJ モデル，平仮名 FT モデル，BCCWJFT モデルの 4 つのモデルを学習し，比較する。本研究における各モデルの作成過程を表した図を図 1 として以下に示す。

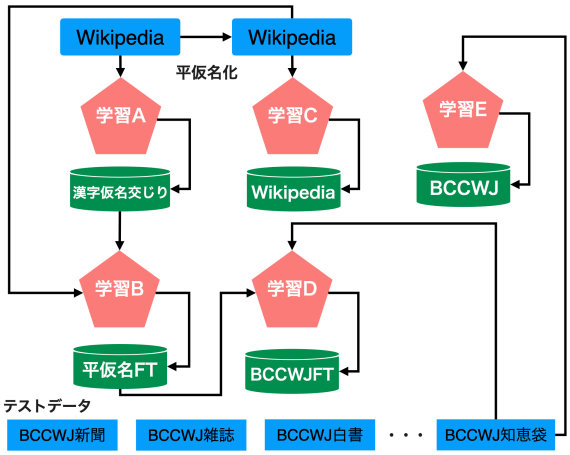

図 1　モデル生成過程

3.1　Wikipedia モデル

Wikipedia モデルは図 1 の学習 C において生成されたモデルである。Wikipedia の漢字仮名交じり文を平仮名に変換したデータを訓練事例として使用する。Wikipedia には平仮名のみのデータがないため，変換には，MeCab の解析結果の読みデータを疑似的な正解として利用した。その際，MeCab の辞書には Unidic を用いた。

3.2　BCCWJ モデル

BCCWJ モデルは図 1 の学習 E において生成されたモデルである。訓練事例としてと Yahoo!知恵袋のデータを使用した。このモデルを使用して Wikipedia のデータと Yahoo!知恵袋のデータの両方を用いて生成されるモデル（図 1 の学習 D）と精度を比較を比較する。

3.3　平仮名 FT モデル

平仮名 FT モデルは，Wikipedia の漢字仮名交じり文を訓練事例として使用したもの（図 1 の学習 A）に対して，Wikipedia の漢字仮名交じり文を平仮名に変換したデータを訓練事例としてファインチューニングを行ったモデル（図 1 の学習 B）である。漢字仮名交じり文には，漢字と平仮名の境目や，漢字の情報があるため，形態素解析の手がかり情報が多い。このことから精度向上が期待できる。

3.4　BCCWJFT モデル

BCCWJFT モデルは図 1 の学習 D において生成されたモデルである。訓練事例として Wikipedia のデータと Yahoo!知恵袋のデータを両方用いることで，精度向上を目的としている。学習をする際の重みの初期値には平仮名 FT モデルを使用し，Yahoo!知恵袋のデータを使ってファインチューニングを行った。

4　実験

4.1　訓練事例

本実験で使用した，訓練事例である Wikipedia のデータは，次の Web サイト

https://dumps.wikimedia.org/jawiki/latest/

において公開されているデータである

jawiki-latest-pages-articles.xml.bz2

を展開し使用した。

訓練事例は学習前にデータを整形した。整形した内容を以下に示す。まず，Wikipedia のデータを MeCab を用いて形態素解析した。MeCab によって出力される結果には素性 (単語分割，品詞，品詞細分類 1，品詞細分類 2，品詞細分類 3，活用型，活用形，基本形，読み，発音) が存在している。表層系の部分を読みに置き換えることにより平仮名のデータを得ることができる。次に，平仮名のデータを 1 文字ずつに分割し，それに対して，品詞をタグとして付与する。ただし，先頭の文字には B-{品詞}が付与し，以降の文字には I-{品詞}を付与した。

例えば，「きみにあう」（君に会う）という平仮名データであれば，以下の表 1 の様な 1 対 1 に対応した文字データとタグを得ることができる。

表 1　「きみにあう」の分割例

き	み	に	あ	う
B-名詞	I-名詞	B-助詞	B-動詞	I-動詞

上記の様にして 1 行ずつのデータを作成し平仮名データとタグのデータをそれぞれファイルに格納することにより，訓練事例を得た。訓練事例の文字数は 1,183,624 個である。

ただし，漢字仮名交じり文を訓練事例として使用したモデルについては，入力は漢字である必要があるので，表層系の部分を読みに置き換える処理は行わなかった。ゆえに，例えば「君に会う」という単

語は平仮名に置き換えないので表 2 の様な 1 対 1 に対応したデータを得ることになる。

表 2　「君に会う」の分割例

君	に	会	う
B-名詞	B-助詞	B-動詞	I-動詞

ただし，BCCWJFT モデルを作成する際に用いる訓練事例は後述する訓練事例である BCCWJ の「Yahoo!知恵袋」のデータセットを使用している。BCCWJ とデータ作成については 4.2 節を参照されたい。

4.2　テストデータ

テストデータとして，国立国語研究所の『現代日本語書き言葉均衡コーパス』[3](以下 BCCWJ と記す)を使用した。

BCCWJ では行形式データが提供されており，本実験では 12 種類のデータセットを使用した。12 種類のデータセットはそれぞれアルファベット 2 文字で提供されている。各アルファベットがどのコーパスであるかと，それぞれのデータセットにおいて本実験で使用した文字数を示す表を表 3 として以下に示す。

表 3　BCCWJ のコーパス対応と文字数

	データセット名	文字数
LB	書籍 (図書館サブコーパス)	1,374,216
OB	ベストセラー	1,093,860
OC	Yahoo!知恵袋	830,960
OL	法律	2,316,374
OM	国会会議録	2,050,400
OP	広報誌	2,151,126
OT	教科書	956,927
OV	韻文	466,878
OW	白書	2,546,307
OY	Yahoo!ブログ	1,305,660
PB	書籍	1,281,251
PN	新聞	1,301,728

本実験では表 3 における 12 種類のコーパスそれぞれ整形し，12 種類の訓練事例を得た。テストデータは訓練事例の作成と同様にして，読みの情報と品詞の情報から平仮名のデータとタグのデータを 1 対 1 に対応する様に作成した。

ただし，「Yahoo!知恵袋」のデータは BCCWJFT モデルを作成する際の訓練事例としても利用している。

3)　https://pj.ninjal.ac.jp/corpus_center/bccwj/

4.3　評価手法

本実験では学習時に，訓練事例の文字データとタグの情報を使用している。学習したモデルは文字データを入力として与えると，モデルが推定する入力に対するタグの情報を出力する。

Wikipedia モデル，平仮名 FT モデル，BCCWJFT モデル，BCCWJ モデルに対して，テストデータを入力として与え，出力されるタグの一致率を正答率として評価する。

また，テストデータの各データセットの評価に対してマクロ平均とマイクロ平均を求めた。

5　実験結果

表 4 に Wikipedia モデル，平仮名 FT モデル，BCCWJFT モデル，BCCWJ モデルの評価結果を示す。表 4 において，Macro はマクロ平均を表し，Micro はマイクロ平均を表している。また，平仮名 FT モデルと BCCWJFT モデルの精度について，アスタリスクが付与されている値は，Wikipedia モデルの精度に対するカイ二乗検定において有意水準 5%で優位であったことを表している。BCCWJ モデルの精度についてアスタリスクが付与されている値は，BCCWJFT モデルの精度に対するカイ二乗検定において有意水準 5%で優位であったことを表している。

また BCCWJFT モデルと BCCWJ モデルにおいて，Yahoo!知恵袋の精度である OC の精度はクローズドデータを用いて評価を行っており，括弧を付与して表示している。また，この 2 つのモデルのマクロ平均とマイクロ平均からは OC の評価データは取り除いている。

表 4 から Wikipedia モデルよりも平仮名 FT モデルの方が精度が高いことが分かる。また，BCCWJFT モデルの精度は平仮名 FT モデルを更にファインチューニングしていることから，Wikipedia モデルよりも精度が向上していることが分かる。

さらに，MeCab を用いてトレーニングデータを評価した。MeCab の辞書は ipadic 辞書の表層系を平仮名に変換したものを使用した。[5] 評価結果はマクロ平均では 79.71%，マイクロ平均では 80.10%となった。ipadic 辞書を使用しており，表 4 の結果とこの結果を単純に比較することができないことに留意する必要がある。

表 4　各モデルに対する評価精度

	Wikipedia	平仮名 FT	BCCWJFT	BCCWJ
LB	56.98	*57.15	*62.76	*63.11
OB	50.14	*50.71	*60.48	*60.16
OC	48.32	*50.23	(*65.50)	(*65.85)
OL	63.39	63.32	*60.55	*60.17
OM	48.49	*49.59	*60.63	*60.49
OP	64.27	*63.93	*64.55	*65.49
OT	57.43	*58.01	*62.04	61.97
OV	41.96	*42.45	*48.82	*47.61
OW	65.83	*65.57	*65.15	*66.89
OY	57.12	57.10	*62.97	*64.02
PB	55.73	55.77	*62.83	*63.28
PN	62.58	*62.30	*64.50	*65.54
Macro	56.02	56.34	61.39	61.70
Micro	58.19	58.40	62.44	62.93

6　考察

表 4 から，Yahoo!知恵袋を訓練事例に用いた BCCWJFT モデルと BCCWJ モデルの精度は，Wikipedia のみを訓練事例に用いた Wikipedia モデルと平仮名 FT モデルよりも高いことが確認できる。このことから，各テストデータに対するモデルの精度はモデルの訓練事例のジャンルによること，つまり Yahoo!知恵袋を訓練事例に用いた方が性能が良いことが分かる。

また，Wikipedia モデルと平仮名 FT モデルを比較すると，平仮名 FT モデルの方がマクロ平均で 0.32 point，マイクロ平均で 0.21 point 精度が向上した。これにより漢字仮名交じりモデルをファインチューニングに利用した平仮名 FT モデルが有効であると言える。

さらに平仮名 FT モデルと BCCWJFT モデルの精度を比較すると，BCCWJFT モデルの方がマクロ平均で 5.50 point，マイクロ平均で 4.04 point 精度が向上した。ゆえに，Wikipedia のデータと BCCWJ のデータを使用したファインチューニングが有効であると言える。

平仮名と BCCWJ 両方のファインチューニングにおいて，Wikipedia モデルの精度が高いテストデータのデータセットに関しては，複数回ファインチューニングを行っても他のデータセットと比べ大幅な精度の向上が見られなかった。これは精度が高いテストデータの文体が Wikipedia の文体に近いからであると推測される。

また，BCCWJFT モデルの精度を BCCWJ モデルと比べると，テストデータのデータセットによって向上しているものと低下しているものがあるものの，マクロ平均とマイクロ平均では BCCWJ モデルの方が精度が高いことが分かる。ジャンルによる差としては，BCCWJFT モデルの方が精度が高いデータセットは，「ベストセラー」「法律」「国会議事録」「教科書」「韻文」の 5 つであり，ジャンルの特徴により精度に差が出ていることが分かる。

最後に，Bi-LSTM CRF モデルの精度は，MeCab のマクロ平均である 79.71 %と比べるとまだまだ改善の余地がある。

7　おわりに

本研究では，Bi-LSTM CRF モデルを用いて平仮名文を単語分割するモデルを作成した。モデルの作成にあたって Wikipedia と Yahoo!知恵袋のデータを平仮名に変換したデータを使ってファインチューニングを行ったところ，Wikipedia だけを訓練事例にしたときよりも，平仮名文の形態素解析の精度が向上すること示した。また，漢字仮名交じり文を訓練事例にして、その後平仮名文を使ってファインチューニングすると，平仮名の形態素解析の精度が向上すること示した。また，Wikipedia と Yahoo!知恵袋のデータを平仮名に変換したデータを使ってファインチューニングを行ったモデルと知恵袋のデータを平仮名に変換したデータだけを使って訓練したモデルの形態素解析の精度の良しあしはジャンルにより異なることを示した。

今後は，平仮名文を分割するモデルの精度をさらに向上させるために，訓練事例数を増やすことを行いたい。加えて，単語の読みや発音，品詞分類といった情報も学習させることでこれらの情報を出力できる様なシステムにしたい。また，本実験では BCCWJFT モデルと BCCWJ モデルの訓練事例に Yahoo!知恵袋のデータセットを使用し精度を測ったが，今後は他のジャンルを使用してモデルを学習し，ジャンルによる精度を差を分析したい。さらに，ドメイン適応の技術を用いて複数のジャンルを組み合わせ全体的な精度を向上させたい。

謝辞

本研究は、茨城大学の特色研究加速イニシアティブ個人研究支援型「自然言語処理、データマイニングに関する研究」に対する研究支援　および JSPS 科研費 17KK0002 の助成を受けたものです。

参考文献

[1]工藤拓, 市川宙, David Talbot, 賀沢秀人. Web 上のひらがな交じり文に頑健な形態素解析. 言語処理学会　第 18 回年次大会　発表論文集, pp. 1272–1275, 2012.

[2]大崎彩葉, 唐口翔平, 大迫拓矢, 佐々木俊哉, 北川善彬, 堺澤勇也, 小町守. Twitter 日本語形態素解析のためのコーパス構築. 言語処理学会　第 22 回年次大会　発表論文集, pp. 16–19, 2016.

[3]藤田早苗, 平博順, 小林哲生, 田中貴秋. 絵本のテキストを対象とした形態素解析. 自然言語処理, Vol. 21, No. 3, pp. 515–539, 2014.

[4]林聖人, 山村毅. ひらがな語の追加と形態素解析の精度についての考察析. 愛知県立大学情報科学部平成 28 年度卒業論文要旨, pp. 1–1, 2017.

[5]井筒順, 明石陸, 加藤涼, 岸野望叶, 小林汰一郎, 金野佑太, 古宮嘉那子. Mecab による平仮名のみの形態素解析. 言語処理学会　第 26 回年次大会　発表論文集, pp. 65–68, 2020.

[6]Ji Ma, Kuzman Ganchev, and David Weiss. State-of-the-art chinese word segmentation with bi-lstms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4902–4908. Association for Computational Linguistics, 2018.

[7]Suphanut Thattinaphanich and Santitham Prom-on. Thai named entity recognition using bi-lstm-crf with word and character representation. In *2019 4th International Conference on Information Technology (InCIT)*, pp. 149–154, 2019.