通時的な領域適応を行った単語分散表現を利用した古文から現代文へのニューラル機械翻訳

高久雅史[1]　平澤寅庄[2]　小町守[2]　古宮嘉那子[1]

[1] 茨城大学　[2] 首都大学東京

16t4027n@vc.ibaraki.ac.jp, hirasawa-tosho@ed.tmu.ac.jp,
komachi@tmu.ac.jp, kanako.komiya.nlp@vc.ibaraki.ac.jp

1　はじめに

本稿では NMT を用いて古文から現代文への翻訳を行う。近年では深層学習と機械翻訳を合わせたニューラル機械翻訳 (NMT) が盛んに研究されている。しかし、古文についての NMT 研究は行われていない。NMT は流暢な出力を出すことが知られていることから、現代文への翻訳の流暢さが上がることが期待される。しかし、一般に NMT は大規模パラレルコーパスを用いてモデル全体を学習させることで実現されるため、小規模パラレルコーパスでの学習では高い翻訳精度のモデルを得ることは難しい [3] と言われている。古文現代文のパラレルコーパスも同様に小規模であり、NMT には適さないと考えられる。

一方、十分な規模のパラレルコーパスがない言語対においては、翻訳モデルの性能を向上させるために、大規模単言語コーパスで事前学習された単語分散表現を使用してモデルを初期化する手法がある [6]。古文においても、同様の手法で翻訳モデルの改善が見込める。

良質な単語分散表現を獲得するためには、規模の大きい学習用コーパスで訓練することが望ましい。そのため、古文に比べより大規模なコーパスから学習された現代語の単語分散表現を利用したい。しかしその一方で、現代文の単語分散表現を翻訳モデルの古文単語の単語埋め込みにそのまま使用すると、ドメインが異なっているため、翻訳性能が落ちることが予想される。もう一つの問題として、古文現代文のパラレルコーパスは、異なる時代の作品データを含む通時コーパスであるため、時代によって単語の意味や形（表層形）が変化するという言語変化の問題がある。

そこで本研究では、現代文の単言語コーパスで学習された単語分散表現に対し、古文の単言語コーパスを使用して時代順に領域適応した後、翻訳モデルを初期化する手法を提案する。提案手法と比較するため、古文の単言語コーパス全部をまとめて、一度のみ fine-tuning した単語分散表現を使用したモデルでも実験を行った。実験の結果から室町時代までは BLEU が改善されるが、平安時代からは低下した。一方で、通時適応を行った翻訳モデルでは、特定の時代にしか現れない語彙を訳出することができ、翻訳品質の改善が見られた。

2　関連研究

星野ら [8] は古文と現代文の段落単位パラレルコーパスから、文分割のためのルールベースのスコア関数を用いて、文単位のパラレルコーパスを得る手法を提案した。それによって得られたパラレルコーパスでの統計的機械翻訳 (SMT) を行い、提案手法による文分割の有効性を示した。古文から現代文への SMT は星野らによって行われたが NMT での研究はされていないため、本研究は初の試みと言える。

言語処理において単語分散表現を用いた研究は盛んに行われている。ニューラル言語処理タスクでも単語分散表現は利用されるが、NMT タスクにおいてはあまり利用されない。これは大規模パラレルコーパスで訓練を行う場合、翻訳モデル自身が適切な単語分散表現を学習するためである。しかし、小規模パラレルコーパスにおける翻訳では、単語分散表現でモデルを初期化することで性能の改善が見込める。Qi ら [6] は小規模なパラレルコーパスしか持たない言語対で NMT モデルを訓練する際に、事前学習済みの単語分散表現を適用することで、翻訳精度が向上することを示した。

単語分散表現の fine-tuning については、Faruqui ら [1] の retrofitting という手法が挙げられる。事前学習された単語分散表現を外部のコーパスで再学習する

ことによって、より良い単語分散表現を得られたことを示した。

また、単語分散表現の通時適応については、Kim ら [2] が通時コーパスを用いて単語分散表現を年ごとにさかのぼる形で学習させる手法を行った。学習によって得られた単語分散表現は、通時コーパスにおける言語変化の検出を可能とした。Kim らの研究から、通時適応させた単語分散表現は、時代による単語の意味変化を捉えることが可能であると考えられる。

3 古文コーパスを用いた単語分散表現の通時的な領域適応

本研究では、翻訳モデルの初期化に用いる単語分散表現を、現代文コーパスからスタートして古文単言語コーパスで新しい時代から古い時代にかけて時代順に fine-tuning する手法を提案する。時代を徐々に戻すのは、適応元の単語分散表現と適応先の単語分散表現との意味のずれが小さくなるようするためである。

単語分散表現の fine-tuning には、Faruqui ら [1] の retrofitting[1] を利用する。事前学習された単語分散表現の時代順 fine-tuning には、近代（江戸以降）・室町・鎌倉・平安・奈良の 5 つの時代に分けた古文コーパスを利用する。単語分散表現の通時適応の流れは以下のようになる。

(1) 事前学習済み単語分散表現を近代のコーパスで fine-tuning する
(2) (1) を室町のコーパスで fine-tuning する
(3) (2) を鎌倉のコーパスで fine-tuning する
(4) (3) を平安のコーパスで fine-tuning する
(5) (4) を奈良のコーパスで fine-tuning する

この手続きで得られた (1)〜(5) の各時代までの単語分散表現を翻訳モデルの単語埋め込み層の初期化に利用する。また、単語埋め込み層のパラメータは固定せず、翻訳モデルの学習によって更新が行われる。

4 実験

4.1 モデル

実験では LSTM をベースとした Attention 付き Encoder-Decoder モデルを使用した。実装にはオープンソースのニューラル機械翻訳ツールである OpenNMT[2] を利用した。ネットワーク構成は、中間層には 2 層の単方向 LSTM を、Attention 層には Global Attention [4] を利用する。Encoder-Decoder ともに、単語ベクトルサイズを 300、中間層の次元数を 512 とした。最適化アルゴリズムには Adam を使用し、学習率は 0.001 にした。学習の際にはモデルが扱う語彙を 20,000 に絞り、それ以外の未知語については<unk>トークンとして処理した。

単語分散表現の初期化は、翻訳モデルの単語埋め込み層の重みの初期値として利用することで実現できる。提案手法では、古文コーパスで通時適応させた事前学習済みの単語分散表現を、翻訳モデル Encoder の単語埋め込み層の重みの初期値に利用した。一方で翻訳モデル Decoder には、事前学習済みの単語分散表現を前処理せずにそのまま初期値として利用した。

モデルの評価には BLEU [5] を用いた。手法ごとに異なるシード値を与え、50,000 step の学習を行う。5,000 step ごとに開発データでの検証を行い、BLEU の値が最高となるモデルでテストした。異なるシードでの学習を 3 回行い、BLEU の平均スコアをモデル全体のスコアとした。

提案手法と比較するため、古文単言語コーパスをまとめて一度のみ fine-tuning を行った際の単語分散表現を利用するモデルでも実験を行った。

4.2 データセット

翻訳では、星野ら [8] が抽出したパラレルコーパスを利用した。このコーパスは、異なる時代の古文作品データが含まれる通時コーパスである。パラレルコーパスの分割割合は先行研究のテストサイズに合わせて (学習:開発:テスト)=(82,591 : 2,000 : 2,093) の分割をした。分割の偏りの影響が少なくなるように、元の 86,684 文対のパラレルコーパス（表 1）は、ランダムサンプリングにて実験データを作成した。コーパスの単語分割には MeCab v0.996[3] を使用し、古文側には中古和文 UniDic v1.3[4]、現代文側には UniDic v2.3.0[4] を辞書として利用した。学習時のモデルへの入力・出力文の長さは 1 文あたり 100 語に制限した。

fine-tuning に用いるコーパスには、国語研究所より提供を受けた小学館新編日本古典文学全集より抽出し

[1] https://github.com/mfaruqui/retrofitting
[2] https://github.com/OpenNMT/OpenNMT
[3] https://taku910.github.io/mecab/
[4] https://unidic.ninjal.ac.jp/

	総文数	語彙サイズ	トークン数
古文	86,684	49,200	2,774,745
現代文		45,690	3,611,783

表 1: 古文現代文パラレルコーパス

	総文数	語彙サイズ	トークン数
近代	22,485	25,584	544,293
室町	12,640	14,931	386,101
鎌倉	35,020	29,062	933,190
平安	59,744	29,520	1,543,102
奈良	4,832	6,013	112,094
計	134,721	61,345	3,518,780

表 2: 古文単言語コーパス

た古文単言語コーパス約 13 万文（表 2）を用いる。古文コーパスの時代分けは、JapanKnowledge の新編日本古典文学全集タイトル一覧[5]を参照した。

4.3 単語分散表現

事前学習済みの単語分散表現には、大規模コーパス NWJC（国語研日本語ウェブコーパス）で学習された、新納ら [7] の nwjc2vec をベースとして利用した。約 258 億語からなるコーパスの NWJC を、形態素解析の辞書には UniDic が利用された。nwjc2vec の fine-tuning はイテレーションを 10 に設定した retrofitting を利用した。古文単言語コーパスは中古和文 UniDic を用いた mecab で単語分割を行った各古文作品データを連結し、fine-tuning のコーパスに用いた。

5 実験結果

表 3 にモデル別の BLEU を示す。“Baseline” は事前学習された単語分散表現を使わず、古文・現代文パラレルコーパスのみで学習した NMT の性能である。また、“SMT” [8] は星野らが提案した古文現代文 SMT の結果の引用である。また、“nwjc2vec” は fine-tuning を用いない nwjc2vec を初期化に利用した結果を指す。

[5] https://japanknowledge.com/contents/koten/title.html

手法	BLEU
SMT [8]	28.02
Baseline	19.43
nwjc2vec	19.11
全古文コーパス	19.83
+アンサンブル	22.11
(1) 近代	19.30
(2) 近代→室町	**19.95**
(3) 近代→室町→鎌倉	19.52
(4) 近代→室町→鎌倉→平安	19.47
(5) 近代→室町→鎌倉→平安→奈良	19.25
+アンサンブル	22.21

表 3: モデル別の BLEU

実験の結果から、特に (2) の室町まで通時適応させた単語分散表現を用いた際に BLEU の改善が大きく見られたが、全古文コーパスをまとめて fine-tuning させた結果と比較して、通時適応による BLEU の改善はあまり見られなかった。また、(1)〜(5) で得られた 5 つのモデルにでのアンサンブル翻訳[6]と、全古文コーパスで fine-tuning された単語分散表現を利用したモデルで精度の高かった上位 5 つを利用したアンサンブル翻訳も行った。アンサンブル翻訳では 2 ポイント以上 BLEU の改善がみられた。

6 考察

BLEU の向上はあまり確認できなかったが、モデルの翻訳結果にはいくつかの効果が見られた。通時適応による効果としては、主に表 4 の例 (1) のような訳出から確認できた。古文の “心に懸る” を現代文では “心にかかる” と訳すのが正解であるが、近代までのコーパスとさらに室町のコーパスまでで fine-tuning させた単語分散表現を利用したモデルでは訳出されなかった。しかし、鎌倉以降のコーパスでは正しく訳出が行えていた。これは、鎌倉のコーパスのみにしか存在しない語彙について、nwjc2vec が領域適応できたと考えられる。

またアンサンブルによる翻訳の精度が高かったことから、2 つのアンサブル翻訳での結果について調べた。

[6] 複数のモデルで同時に予測を行った結果を組み合わせて、確率的に出力を決定する方法

(1) 鎌倉のコーパスによる通時適応で翻訳が改善された例	
入力文:	時宗 は 親 の 敵 より 外 に 、 心 に 懸る こと 候は ず 。
参照文:	時宗 は 親 の 敵 以外 に は 、 心 に かかる こと は あり ませ ん 。
Baseline:	私 は 親 の 敵 より ほか に 、 心 に <unk> こと は あり ませ ん 。
近代→室町:	時宗 は 親 の 敵 より ほか に ある 。
近代→室町→鎌倉:	私 は 親 の 敵 より ほか に 、 心 に かかる こと は あり ませ ん 。
(2) 提案手法のアンサンブル翻訳で翻訳が改善された例	
入力文:	いと さわがしき まで あれ ど 、 とが も なし 。
参照文:	まこと に 騒がしい ほど で ある けれど 、 それ は 別 に 、 困る と いう こと も ない 。
Baseline:	まこと に <unk> まで も ある けれど 、 お 咎め も ない 。
全古文アンサンブル:	ひどく <unk> まで も ある けれど 、 <unk> も ない 。
通時適応アンサンブル:	まこと に 騒がしい くらい に なっ て いる けれど も 、 <unk> も ない 。

表 4: 翻訳のサンプル

それぞれの翻訳結果の語彙集合を調べた所、提案手法での翻訳結果が 4,476 異なり語、全時代での結果が 4,468 異なり語であり、わずかではあるが提案手法の方が多くの語彙を訳出している。実際の例では、(2) の例文が確認できた。全古文コーパスでの fine-tuning では “騒がしい” という語が拾えていないが、通時適応で訳出が行えていたことが確認できた。

7 おわりに

本研究では、古文現代文による初の NMT の試みと、単語分散表現の利用時に段階的な fine-tuning をした単語分散表現の初期化手法について提案した。実験の結果、通時適応した単語分散表現の利用によって、時代固有の語の訳出を可能としたり、他のモデルよりも多くの語彙を訳出できたことが確認された。一方で、時代によってはモデルがあまり改善されなかった。今後の研究としては、翻訳対象の時代に合わせた通時適応の効果の調査が必要である。特に、ある時代までで fine-tuning した単語分散表現をその時代のパラレルコーパスの翻訳に利用して、どれくらいモデルが改善されるかを明らかにする必要がある。また、nwjc2vec ではない単語分散表現をベースとしたり、通時適応でモデルの改善がよく見られた時代（本稿では室町のコーパス）に絞った fine-tuning の効果についても調べる。

謝辞

本研究の一部は国立国語研究所の共同研究プロジェクト「通時コーパスの構築と日本語史研究の新展開」の研究成果を報告したものです。また、富士通研究所の横野光様には、先行研究のデータをいただきました。御礼申し上げます。

参考文献

[1] Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. Retrofitting word vectors to semantic lexicons. In *NAACL*, 2015.

[2] Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. Temporal analysis of language through neural language models. In *ACL*, 2014.

[3] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *ACL*, 2017.

[4] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[5] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[6] Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. When and why are pre-trained word embeddings useful for neural machine translation? In *NAACL-HLT*, 2018.

[7] 新納浩幸, 浅原正幸, 古宮嘉那子, 佐々木稔. nwjc2vec: 国語研日本語ウェブコーパスから構築した単語の分散表現データ. 自然言語処理, Vol. 24, No. 5, 2017.

[8] 星野翔, 宮尾祐介, 大橋駿介, 相澤彰子, 横野光. 対照コーパスを用いた古文の現代語機械翻訳. 言語処理学会第 20 回年次大会, 2014.