JParaCrawl を用いた単文翻訳におけるドメイン適応

名村太一 [1] 村上仁一 [2]
[1] 鳥取大学工学部電気情報系学科
[2] 鳥取大学工学部
[1]B19T2076C@edu.tottori-u.ac.jp
[2]murakami@tottori-u.ac.jp

概要

NMT において様々なドメイン適応 (domain adaptation) の手法が研究されている [1]．また，ドメイン適応のためには大規模なコーパスが必要である．そこで大規模コーパスとして JParaCrawl[2] が開発された．しかし，まだ人手の翻訳精度に及ばない．本研究ではファインチューニングに用いるコーパスが小さいことが原因だと考える．そこで JParaCrawl からテスト文に類似する対訳を抽出し，コーパスを拡張しファインチューニングを行う．実験の結果，対訳を追加しなかった場合と比較して，BLEU 値が 1.7 向上ｓした．

1 はじめに

NMT においてドメイン適応についての様々な研究がされてきた [1]．また，ドメイン適応を行うためには十分なコーパスが必要である．しかし，日英対訳コーパスにおいては一般的なドメインの大規模コーパスが無かった．そこでウェブから大規模かつ特定のドメインに特化しない日英対訳コーパスとして JParaCrawl[2] が開発された．しかし，まだ人手の翻訳精度に及ばない．

本研究ではファインチューニングに用いるコーパスが小さいことが原因だと考える．そこで，ファインチューニングに用いるコーパスを拡張することで翻訳精度の向上を図る．まず，目的ドメインのテスト文に類似する対訳を JParaCrawl から抽出する．次に，抽出したコーパスと目的ドメインのコーパスを合わせた新たなコーパスを作成する．そして新たなコーパスを用いて JParaCrawl の事前学習済みモデルのファインチューニングを行う．

また，目的ドメインのコーパスのみを用いて JParaCrawl の事前学習済みモデルをファインチューニングしたものを BaseLine とする．BaseLine と提案手法の比較を，BLEU 値による自動評価と人手による対比較評価で行う．

2 過去研究

2.1 JParaCrawl

NTT コミュニケーション科学基礎研究所によって作成された日英対訳コーパスである．対訳文は約 2000 万文が用意されている．

2.2 ドメイン適応

ドメイン特有の言い回しや表現がある．そのため NMT において，汎用的なシステムよりそのドメインに適応したシステムの方が翻訳精度が向上する [4]．ドメイン適応するために様々な手法が研究されているが，コーパス中心とした適応 (data centric) と，モデル中心とした適応 (model centric) に大きく分けられる [1]．本研究ではこれらを組み合わせてドメイン適応を行う．

2.2.1 データ選択

データ選択 (data selection) とはコーパスによるドメイン適応の手法である [1]．目的ドメインのコーパスに類似する文を他のコーパスから抽出し，コーパスを拡張する．様々なアルゴリズムが考案されているが [5]，本研究では TF-IDF を用いる．

2.2.2 ファインチューニング

ファインチューニング (fine-tuning) とはドメイン適応の手法である [6][1]．事前学習済みモデルを目的ドメインのコーパスを用いてパラメータを微調整することである．これによってモデルをドメインに適応させる．本研究では JParaCrawl のファインチューニングの方法に従って行う．

3 提案手法

本研究では，以下の手順を提案する．

1. TF-IDF でテスト文に類似する対訳を JParaCrawl から抽出
2. 抽出した対訳を目的ドメインのコーパスに付け加えて，新たなコーパスを作成
3. 新たなコーパスを用いて，JParaCrawl で学習された事前学習済みモデルをファインチューニング

4 実験

4.1 実験データ

実験に用いるデータを表 1 に表す. 単文対訳コーパスとテスト文は電子辞書を中心に採取された単文を用いる [7]. また，本実験では JParaCrawl の対訳の内，Bicleaner の値を 0.7 以上の対訳のみを使用する.

表 1 実験データ

JParaCrawl ver 3.0 (Bicleaner 0.7 以上)	18,450,971
単文対訳コーパス	163,188
テスト文	16,328

4.2 実験設定

FAIRSEQ[8] を用いて実験を行う. 学習のパラメータは [2] に従って，同様の設定で行う. そのため，事前学習済みモデルから 2,，000 ステップの学習を行うことでファインチューニングを行う．また，前処理には sentencepiece[9] を用いて，語彙数 32,000 とし，unigram 単位で分割する.

4.3 実験手順

1. TF-IDF でテスト文に類似する文を JParaCrawl から抽出する
2. 単文対訳コーパスと抽出した文を合わせたコーパスを用いて，JParaCrawl で学習された事前学習済みモデルをファインチューニングする
3. テスト文の翻訳を行い，評価を行う

抽出する量を，テスト文 1 文に対して類似した上位 10 文を Proposed として実験を行う.

評価には SacreBLEU[10] を用いて，BLEU[11] を計算する. また，人手による対比較評価も行う.

4.4 BaseLine

本実験では，BaseLine を単文対訳コーパスのみを用いて事前学習済みモデルをファインチューニングした結果とする.

5 実験結果

5.1 自動評価

表 2 に BLEU 値の評価結果を示す.

表 2 提案手法と BaseLine の BLEU 値

model	コーパス量	BLEU
Proposed	324,233	**26.2**
BaseLine	163,188	24.5

表 2 より，BLEU 値において Proposed は BaseLine より向上することが確認できた.

5.2 人手評価

人手評価は同研究室の学生 4 名に行ってもらった．そしてその平均値を人手評価とする.

Proposed と BaseLine の対比較評価を行う. テスト文の中からランダム 100 文に対して，Proposed の方が良い，BaseLine の方が良い，ほとんど差がないの 3 つの分類で対比較評価を行った. その結果を表 3 に示す.

表 3 Proposed と BaseLine の対比較評価

評価	割合 (%)
Proposed	28.5
BaseLine	16.8
差がない	54.8

その結果，BaseLine の方が良いが 16.8%に対して，Proposed の方が良いが 28.5%で上回った. 人手評価でも Proposed が BaseLine を上回る結果になった.

5.3 出力例

5.3.1 Proposed の良い例

BaseLine の翻訳結果と比較して Proposed の方が良い文を，追加された文とともに表 4 に示す.

表 4 より，BaseLine では「知床」が翻訳されなかったのに対して，Proposed では「知床」が含まれる文が追加されることによって，正しく翻訳されるようになった.

5.3.2 Proposed が悪かった例

BaseLine の翻訳結果と比較して Proposed の方が悪かった翻訳と，追加された文を表 5 に示す.

「葬られた」の翻訳は「hushed up」が正しいが，「葬られた」と「buried」を含む対訳を学習したため，間違った翻訳をしたと考えられる.

表 4　Proposed の良い翻訳例と，追加された文

	文
入力文	知床では自然破壊が進んでいる.
Proposed	Natural destruction is progressing in Shiretoko .
BaseLine	In the intellectual world , destruction of nature is increasing .
参照文	The destruction of the natural environment around Shiretoko has become serious .
類似文 1 日本語文	知床自然の宝庫知床は，生物と自然が共有した世界を見ることができます.
類似文 1 英語文	Shiretoko Treasury of nature Shiretoko can see the world shared by living things and nature.
類似文 2 日本語文	冬の知床世界自然遺産として知られる知床.
類似文 2 英語文	Winter's Shiretoko Shiretoko known as a natural world heritage site.

表 5　Proposed の悪い翻訳例と，追加された文

	文
入力文	その事件はうやむやに葬られた.
Proposed	The case was buried peacefully .
BaseLine	The matter was hushed up .
参照文	The matter has been hushed up .
類似文 1 日本語文	遺体は後に，裸にされて傷つけられた状態で発見され，ブロードフットの遺体とともにラグマーニーにある私の兄の要塞の近くに葬られた.
類似文 1 英語文	The body was afterwards found, naked and mutilated, and, with Broadfoot's, was buried near my brother's fort Lughmaunee.

6　考察

6.1　抽出するデータ量を変更

6.1.1　自動評価結果

この節では JParaCrawl から抽出するデータ量を変更する．Proposed ではテスト文 1 文に対して類似する上位 10 文を抽出していた. ここではテスト文 1 文に対して類似する上位 1 文と 100 文としたものを top1 と top100 として実験を行う.

表 6　BaseLine と top1，top100 の BLEU 値

model	コーパス量	BLEU
top1	179,501	25.0
top100	1,559,702	24.0
BaseLine	163,188	24.5

表 6 より，top1 では BaseLine より BLEU 値が向上した. Proposed の方が top1 と比べて BLEU 値が高かった. また，top100 では BaseLine より BLEU 値が低下した.

6.1.2　Proposed と top1 の違いの例

top1 の結果から BaseLine よりも BLEU 値が高いため，Proposed と比較したものを表 7 に示す.

表 7　top1，Proposed，BaseLine の翻訳を比較

	文
入力文	私は酒で頭がぼんやりしてしまった。
top1	My brain has been muddled by alcohol .
Proposed	My head went blank from drinking .
BaseLine	My brain has gotten muddled with alcohol .
参照文	The spirits muddled my brain .
Proposed 類似文 日本語	持続時間も長く、頭が真っ白になってしまうという声も。
Proposed 類似文 英語	Some people say that it lasts for a long time and that it makes them go blank.

この結果では，BaseLine より top1 の方がよくなった．しかし，JParaCrawl より抽出する量を増やした Proposed はかえって悪くなった．この原因として，top1 には無かったが Proposed には含まれていた類似文が影響したと考える.

6.2　ランダムに追加

この節では JParaCrawl からランダムに抽出した対訳を追加し，学習を行う．それぞれ 1 万，10 万，100 万の対訳を抽出し，追加したコーパスで学習を行った. その結果を表 8 に示す.

表 8　BaseLine と random に 1 万，10 万，100 万を追加して学習した BLEU 値

model	コーパス量	BLEU
random 10,000	173,188	24.7
random 100,000	263,188	25.7
random 1,000,000	1,163,188	23.6
BaseLine	163,188	24.5

random 100,000 が最も BLEU 値が高い結果だった．しかし，類似文を追加した Proposed の方が BLEU 値が向上した.

6.3　抽出した対訳のみでファインチューニング

この節では単文対訳コーパスを使用せずに，JParaCrawl より抽出したコーパスのみでファインチューニングを行う．抽出する量をテスト文 1 文に対して 1 文，10 文，100 文とし，それぞれ JPara top1，JPara top10，JPara top100 とする．その結果を表 9 に示す.

表 9 より，全ての結果で BaseLine を下回ったが，類似する文を増やすと BLEU 値が向上した．一方表

表 9 JParaCrawl からテスト文に類似する文を抽出したコーパスのみでファインチューニングを行ったモデルの BLEU 値

model	コーパス量	BLEU
JPara top1	16,313	14.6
JPara top10	161,045	17.8
JPara top100	1,396,514	20.5
BaseLine	163,188	24.5

6 では，top100 において，コーパスを大きくしすぎると BLEU 値が低下した．しかし，この実験では BLEU 値が向上した．

6.4 JParaCrawl+単文対訳コーパスから抽出

テスト文により類似したコーパスでファインチューニングを行えば翻訳精度が向上すると考える．そのため，この節では単文対訳コーパスと JParaCrawl を合わせたコーパスからテスト文に類似する文を抽出する．抽出する量は，テスト文 1 文に対して 1 文，10 文，100 文とし，それぞれを JPara+単文 top1，JPara+単文 top10，JPara+単文 top100 とする．その結果を表 10 に示す．

表 10 単文対訳コーパス+JParaCrawl からテスト文に類似する文を抽出したコーパスでファインチューニングを行ったモデルの BLEU 値

model	コーパス量	BLEU
JPara+単文 top1	16,130	16.9
JPara+単文 top10	155,781	21.2
JPara+単文 top100	1,279,893	22.8
BaseLine	163188	24.5

表 10 より，全ての結果で BaseLine を下回ったが，類似する文を増やすと BLEU 値が向上した．また，表 9 と比較して全体的に BLEU 値が向上した．

この結果より，テスト文に類似していることより一定量の単文対訳コーパスが重要であると考える．単文対訳コーパスは単文の短い文で構成されている．それに対して JParaCrawl の多くは長い文で構成される．この違いにより表 2 の結果が最も良かったと考える．

6.5 Google 翻訳との比較

Google 翻訳との比較を行う．表 11 で Google 翻訳と BLEU 値で比較する．また，表 12 に，Proposed と Google 翻訳の対比較評価の結果を示す．

表 11 Google 翻訳と BLEU 値で比較

model	BLEU
Proposed	26.2
Google	26.9
BaseLine	24.5

表 12 Proposed と Google 翻訳の対比較評価

評価	割合 (%)
Proposed	13.3
Google	33.0
差がない	53.8

表 13，14 に Google 翻訳と Proposed の例を示す．

表 13 Proposed と Google 翻訳を比較して，Proposed が良かった例

	文
入力文	雨で試合が流れた。
Proposed	The game was rained out .
Google	The game was washed away by the rain.
参照文	The game was rained out .

表 14 Proposed と Google 翻訳を比較して，Google が良かった例

	文
入力文	彼が得た報酬は微々たるものだった。
Proposed	His remuneration he received was substantive .
Google	His reward was meager.
参照文	He only got a small reward .

表 11 より，BLEU 値において，Proposed の値は Google 翻訳の値に近かった．しかし，表 12 の結果ではまだ及ばなかった．

7 おわりに

本研究では，TF-IDF を用いて JParaCrawl からテスト文に類似する対訳文を抽出した．その対訳文を目的ドメインのコーパスに追加することでコーパスを拡張した．拡張したコーパスを用いて事前学習済みモデルのファインチューニングを行うことで BaseLine と比較して BLEU 値と人手評価の両方で翻訳精度の向上が見られた．具体的には BLEU で 1.7 向上した．

コーパスを拡張しすぎると翻訳精度の低下が確認されたので，適切なコーパスの量や，他の対訳の抽出法を今後の課題にする．

謝辞

人手評価に協力してくれた，以下の 5 名の学生の協力を得ました．感謝致します．（柳原 弘哉，丸山 京祐，深谷 諒，松本 武尊）

参考文献

[1] Chenhui Chu and Rui Wang. A survey of domain adaptation for neural machine translation. In **Proceedings of the 27th International Conference on Computational Linguistics**, pp. 1304–1319, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[2] 森下睦, 帖佐克己, 鈴木潤, 永田昌明. Jparacrawl v3.0: 大規模日英対訳コーパス. 言語処理学会 第 28 回年次大会 発表論文集 (2022 年 3 月）, 2022.

[3] Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. Paracrawl: Web-scale acquisition of parallel corpora. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 4555–4567, 2020.

[4] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. **arXiv preprint arXiv:1706.03872**, 2017.

[5] Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. Transductive data-selection algorithms for fine-tuning neural machine translation. **arXiv preprint arXiv:1908.09532**, 2019.

[6] Minh-Thang Luong and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In **Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign**, 2015.

[7] 村上仁一. 日英対訳データベースの作成のための 1 考察. 言語処理学会第 17 回年次大会発表論文集, D4-5, pp. 979–82, 2011.

[8] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. **arXiv preprint arXiv:1904.01038**, 2019.

[9] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. **arXiv preprint arXiv:1808.06226**, 2018.

[10] Matt Post. A call for clarity in reporting BLEU scores. In **Proceedings of the Third Conference on Machine Translation: Research Papers**, pp. 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics.

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In **Proceedings of the 40th annual meeting of the Association for Computational Linguistics**, pp. 311–318, 2002.