クラウドソーシングによる Web サイトマイニングを用いた翻訳モデルの即時領域適応

森下 睦 [1,2], 鈴木 潤 [2], 永田 昌明 [1]
[1]NTT コミュニケーション科学基礎研究所　[2] 東北大学
{makoto.morishita.gr, masaaki.nagata.et}@hco.ntt.co.jp
jun.suzuki@ecei.tohoku.ac.jp

1　はじめに

2020 年は新型コロナウイルス感染症の流行により、正しい情報を迅速に収集し行動することが求められた。しかし、日本国内においては政府や自治体からの情報の大半は日本語により発信されたため、日本国内に滞在する非日本語話者にとっては最新の情報を受け取りにくい状況が生じていた。機械翻訳を用いて日本語を多言語に翻訳するケースも見受けられたが、特殊な用語が多く含まれる分野ゆえに十分な翻訳品質が得られない場合があった [1]。

このような状況で十分な翻訳品質が得られない原因の一つは翻訳モデルの学習データにあると考えられる。近年のニューラル機械翻訳モデルの多くは対訳コーパスをもとに学習を行う。この際、翻訳対象領域に関する対訳コーパスが十分に学習していない場合、翻訳品質が低下する欠点がある [2]。そのため、必要十分な翻訳品質を達成するためには、翻訳したい領域 (分野) の対訳コーパスを十分に用意し、適用領域に特化した翻訳モデルを構築するのが、最も有効な手段だと考えられる。しかし、一般に存在する対訳コーパスは限定的であり、特化したい領域の対訳文を十分に用意できない場合多い。

本研究では、特定領域に特化した機械翻訳モデルが必要となった際に、既存の翻訳モデルを迅速かつ安価に適応する手法を検討する。具体的には、図 1 のようにクラウドワーカーから特定領域に関連する対訳となっている URL 対の提供を受け、翻訳対象領域の対訳コーパスを迅速に Web から構築し、翻訳モデルを適応することで高い翻訳品質を達成する。本手法により、前述のような緊急事態発生時にも数日以内に高い翻訳品質を持った翻訳器を提供することを目標とする。

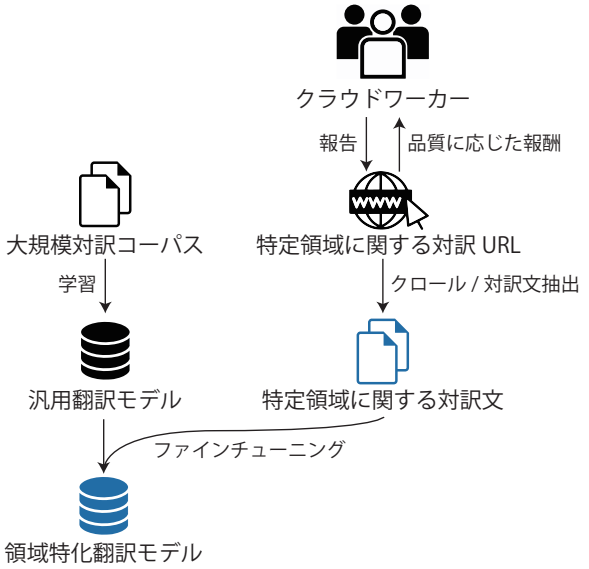

図 1　クラウドソーシングを用いた領域適応の概要図

2　関連研究

2.1　翻訳モデルの領域適応

翻訳対象領域が限定されている際に、翻訳モデルを特定領域に特化させることを領域適応という。近年主流のニューラル機械翻訳では、大規模対訳コーパスで事前に学習した翻訳モデルを、適応対象領域の対訳コーパスを用いてファインチューニングする手法が主流となっている。Kiyono らは WMT2020 シェアードタスクにおいて、本手法により翻訳モデルをニュース領域に適応させることで、最大 2.2 ポイントの BLEU スコア向上を記録した [3]。ただし、本手法は適応対象領域のみを含んだ対訳コーパスが存在する場合に使用可能である。一般的には、特定領域のみを含んだ対訳コーパスが存在することは稀なため、大規模な対訳コーパスから適応対象領域に近い文を抽出し領域適応を行う場合が多い [4]。

本研究では、適応対象領域に近い対訳文を Web から収集し、得られた対訳文を用いて翻訳モデルをファインチューニングすることで領域適応を実現する。これまでの手法は、既存の大規模対訳コーパスから適応対象領域に近い対訳文を抽出し学習するものが大半だったのに対し、本手法では対訳文を新たに Web から収集するため、既存の対訳コーパスが網羅していない分野についても適応可能である。

2.2 Web からの対訳文収集

近年は Web から対訳文を抽出し、大規模な対訳コーパスを構築する研究が進んでいる。ParaCrawl プロジェクトは、Web をもとに EU 公式言語-英語間の大規模な対訳コーパスを作成している [5]。また、日本語-英語間では 1000 万文対を超える Web ベース大規模対訳コーパス JParaCrawl が公開されている [6]。これらのコーパスは、CommonCrawl から各 Web サイトに含まれる言語別データ量を分析し、収集対象言語対のデータが一定以上の Web サイトをクロール対象とすることで構築されている。この手法では、膨大な Web から効率的に対訳文を収集することができるが、目的言語対のデータ量がしきい値未満の Web サイトについては取りこぼしが発生するほか、CommonCrawl に含まれていない Web サイトについては収集対象とならない欠点があった。さらに、本研究のように翻訳したい領域があらかじめ決まっている状態でも、それに合わせた対訳文を収集することはできなかった。

Ling らはクラウドソーシングを活用して Twitter から対訳文を抽出する手法を提案した [7]。この手法では、安価に対訳文を収集することができるものの、特定領域に特化した対訳文は収集できない。

また Papavassiliou らは、Web クローラを活用することで特定領域の対訳コーパスを構築する手法を提案した [8]。本手法は、特定領域に関する対訳文を収集できるものの、Web クローラを動作させるためには時間とコストがかかる。

以上の研究をもとに、本研究では特定領域の対訳文を Web から迅速かつ安価に収集する手法を検討する。これにより、得られた対訳文を用いて特定領域に特化した翻訳モデルが学習可能となる。

3 特定領域対訳文収集

3.1 クラウドソーシングによる対訳 URL 収集

図 1 に示したように、本研究ではクラウドソーシングを活用して特定領域対訳文を収集し、領域特化翻訳モデルの実現を目指す。具体的な手順は以下の通りである。まず、適応対象領域に関する少量の対訳コーパス (~1000 文程度) を人手で用意しこれを Dev セットとする。クラウドワーカーへ Dev セットを提示し、Dev セットに近い対訳文を含む Web ページ URL 対の報告を依頼する。クラウドワーカーへは発見した URL1 対につき、規定の報酬を支払う[1)] (3.3 節参照)。

本研究では、クラウドワーカーはこれまでの経験や勘を活かし、対訳となっている Web ページを探し出せると仮定している。この仮定が成り立てば、本手法によりこれまでは難しかった特定領域に特化した対訳文収集が可能になる。

3.2 対訳文抽出

クラウドワーカーから対訳となっている URL 対の報告を受けると、自動的に対訳文抽出を行う。ParaCrawl 等の先行研究では、文書対応付けを行い Web サイト全体からどのページ同士が対応しているかを求め、その後対訳文対応付けを行っていた。しかし、本研究ではクラウドワーカーから対応しているページの URL 対を受け取るため、文書対応付けは必要ない。対訳文対応付けには、`vecalign` を用いる [9]。`vecalign` は `LASER` [10] の出力である多言語文埋め込みを受け取り、対訳文を抽出する。本研究では、`vecalign` が出力するスコア 0.5 以上の対訳文のみを正しい対訳になっていると仮定して以降の実験で使用する。

3.3 報酬設定

クラウドワーカーへ支払う報酬は、報告された対訳 URL から得られた対訳文数、対訳品質および適応対象領域への類似度により変化させる。これにより、クラウドワーカーはより翻訳品質向上に寄与する Web ページを探索する意欲が増すことが期待される。また、不適切な作業については報酬を低く抑えることができ、コストの削減にも繋がる。実際の

1) ただし、すでに報告されている URL との重複は許さない。

報酬額は以下のように決定される。

対訳品質スコア　報告された対訳 URL から得られた対訳文を $\mathbb{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ とする。ここで、x_i は i 番目の原言語文、y_i は i 番目の目的言語文を表す。対訳品質スコア S_a は vecalign によって算出された 0 から 1 の値を持つスコア $\mathrm{V}(x, y)$ を用いて以下のように計算される。

$$S_a = \sum_{(x_i, y_i) \in \mathbb{D}} \mathrm{V}(x_i, y_i) \tag{1}$$

領域類似度スコア　抽出された原言語文全体の平均文埋め込みを求め、これを Web ページ全体の文書埋め込み $\boldsymbol{e}$ とする。

$$\boldsymbol{e} = \frac{1}{|\mathbb{D}|} \sum_{i=1}^{|\mathbb{D}|} \mathrm{L}(x_i) \tag{2}$$

ここで、$\mathrm{L}(x)$ は文 x について LASER による多言語文埋め込みを返す関数、$|\mathbb{D}|$ は抽出された対訳文数を表す。

本研究では、適応領域に関する Dev セットは文書単位に分割されていることを仮定する。m 文書からなる Dev セットに関して、文書埋め込みを同様に計算する: $\boldsymbol{E}_d = (\boldsymbol{e}_{d_1}, \ldots, \boldsymbol{e}_{d_m})$。

領域類似度スコア S_d は、Web ページ文書埋め込み $\boldsymbol{e}$ と Dev セット文書埋め込み $\boldsymbol{E}_d$ の cos 類似度を用いて以下のように計算する。

$$S_d = |\mathbb{D}| \max_{\boldsymbol{e}_d \in \boldsymbol{E}_d} \frac{\boldsymbol{e} \cdot \boldsymbol{e}_d}{\|\boldsymbol{e}\| \|\boldsymbol{e}_d\|} \tag{3}$$

最終的な報酬額 r は、上記のスコアをもとに最低保証金額である 20 円を加え、$r = 20 + 3S_a + 3S_d$ 円とした。ただし、報告された URL 対から 1 文も対訳文が得られなかった場合は報酬は 0 円となる。

クラウドワーカーに対しては、各作業について報酬及び各種スコアをフィードバックする。これにより、ワーカーはより高い報酬を得るよう、より品質が上がるように作業内容を変化させることが期待される。

4　実験

4.1　実験設定

対訳文収集　本実験では、適応対象領域として新型コロナウイルス感染症を対象とした。本領域に関する対訳コーパス構築のために、クラウドワーカーを 10 人募集し、5 日間かけて対訳文収集を実施した。報酬額の設定は 3.3 節の通りである。

表 1　Dev/Test セットに含まれる文数および英語側単語数

	文数	単語数
Dev	686	14,190
Test	1,820	43,272

表 2　BLEU スコア

	Dev	Test
JParaCrawl のみ	25.9	33.0
Moore-Lewis を用いた領域適応	25.3	33.1
提案法を用いた領域適応	**27.3**	**34.5**

表 3　分野別 BLEU スコア

分野	JParaCrawl のみ	領域適応モデル
医療会話	15.3	**16.7**
医学論文	37.8	**38.8**
ニュース	29.4	**33.4**
Wikipedia	30.9	**32.8**
告知文	**26.4**	24.4

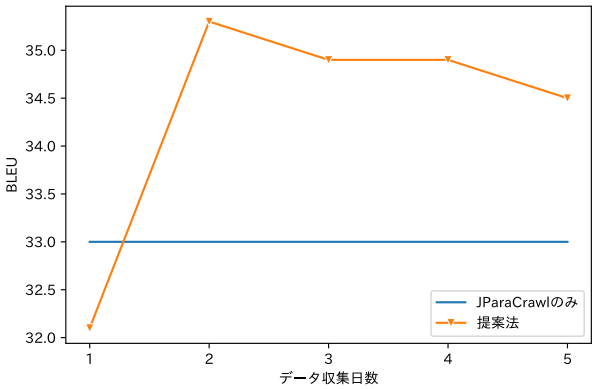

図 2　データ収集日数による BLEU スコアの変化

翻訳モデル　本実験では、英日翻訳を対象とした。分野適応前の汎用翻訳モデルとして、約 1000 万文の日英対訳コーパス JParaCrawl v2.0 で学習した Transformer Big モデル [11] を使用した。JParaCrawl は新型コロナウイルス感染症流行前に収集されたものであるため、対象領域に関する対訳文は含んでいない。汎用翻訳モデルの学習に関するハイパーパラメータは Morishita らの設定と同一である [6]。

本実験では、汎用翻訳モデルをもとに、提案法によりクラウドソーシングを活用して収集した対訳文を用いてモデルをファインチューニングすることで領域適応を行った。また、領域適応のベースラインとして、既存の大規模対訳コーパス (JParaCrawl) から特定領域に近い文を抽出し学習する Moore-Lewis の手法を使用する [12]。本手法は、2010 年に提案された比較的古い手法ではあるものの、2020 年に提案

原文	Japan's National Center for Global Health and Medicine (NCGM) is planning a clinical trial for Teijin's Alvesco (ciclesonide), an inhaled corticosteroid for asthma, for the treatment of pre-symptomatic patients infected with the novel coronavirus.
参照訳	日本の国立国際医療研究センター（NCGM）は、新型コロナウイルスに感染した症状が出る前の患者の治療に、喘息用の吸入型コルチコステロイドである帝人社のオルベスコ（シクレソニド）の臨床研究を計画している。
JParaCrawl のみ	国立国際医療研究センター (NCGM) は、帝人の喘息用コルチコステロイド「アルベスコ (シクロソニド)」の新規コロナウイルス感染前症状患者を対象とした臨床試験を計画している。
領域適応モデル	国立国際医療研究センター (National Center for Global Health and Medicine: NCGM) は、帝人の喘息用コルチコステロイドを吸入したアルベスコ (シクロソニド) について、新型コロナウイルスに感染した症状前の患者の治療のための臨床試験を計画している。

図 3　翻訳品質が改善した例

された手法とほぼ同等の性能をもった強力なベースラインとなっている [13]。ファインチューニング時のハイパーパラメータを付録 表 4 に示す。

適応対象領域に関する Dev セットおよび Test セットとして、新型コロナウイルス感染症に関する多言語テストセットである TICO-19 [14] を日本語訳し使用した。使用した Dev セットおよび Test セットの文数、単語数を表 1 に示す[2)]。

4.2　実験結果/考察

5 日間のデータ収集の結果、提案法により得られた対訳文数は 6,772 文、総報酬額は 31,079 円となった。得られた対訳文をもとに汎用翻訳モデルをファインチューニングした際の BLEU スコアを表 2 に示す[3)]。既存の Moore-Lewis 法では汎用翻訳モデルとほぼ同等の品質に留まったが、提案法により収集した対訳文を用いた領域適応は 1.5 ポイントの BLEU スコア向上を達成した。これは、既存の対訳コーパス中に適応領域に近い対訳文が十分に含まれていなかったことが原因だと考えられる。また、提案法による BLEU スコア向上はクラウドワーカーが適切に適応領域に近い対訳 URL を探索できたことを示している。

図 2 に、データ収集日数による BLEU スコアの変化を示す。収集 1 日目は収集できた対訳文数が少ない (1,152 文) こともありベースラインモデルを下回っているが、2 日目にはベースラインを大幅に上回る翻訳品質を達成している。

TICO-19 に含まれる分野別 BLEU スコアを表 3 に示す。提案法による領域適応モデルは、ほとんどの分野でベースラインモデルを上回るスコアを達成している。特に、ニュース分野に関してはベースラインモデルと比較して 4 ポイントの BLEU スコア向上が見られる。これは、新型コロナウイルス感染症に関連したニュースサイトがクラウドワーカーによって多く報告されたことに起因すると思われる。

提案法により翻訳品質が改善した例を図 3 に示す。JParaCrawl のみを用いて学習したモデルでは "novel coronavirus" を正しく翻訳できていないのに対し、提案法による領域適応を行ったモデルでは「新型コロナウイルス」と正しく翻訳できていることがわかる。このように、適応対象領域の対訳文を新たに収集することにより、適応領域に関する専門用語が正しく翻訳可能になるケースが多く見受けられた。

以上の実験から、提案法により、特定領域の翻訳品質を迅速かつ安価に向上させられることを示した。

5　おわりに

本稿では、クラウドワーカーを活用した翻訳モデルの即時適応手法を提案した。新型コロナウイルス感染症に関する領域を対象に行なった実験により、提案法を用いることで既存の対訳コーパスが網羅していない領域に対しても迅速かつ安価に領域適応を行うことができることを示した。今後、日英以外の言語対や、新型コロナウイルス感染症以外の様々な領域に対しても適応可能であることを示していきたい。

2)　翻訳作業期間の都合上、Dev セットおよび Test セットの一部のみを翻訳し使用する。そのため、本来の TICO-19 全体に含まれる文数および単語数と差が生じている。

3)　Moore-Lewis 法では、使用する対訳文数を変えて複数のモデルを学習し最も Dev セットの BLEU が高かかったモデル (上位 4,000 文) を使用した。試行した対訳文数は次の通り: スコア上位 1,000 文, 2,000 文, 4,000 文, 6,000 文, 10,000 文, 25,000 文, 50,000 文, 100,000 文。

参考文献

[1] 厚労省 HP、新型肺炎の外国語情報で誤訳多発 「手洗い重要」が「トイレ重要」. 毎日新聞.

[2] Mathias Müller, Annette Rios, and Rico Sennrich. Domain robustness in neural machine translation. In Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track), pages 151–164, 2020.

[3] Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. Tohoku-AIP-NTT at WMT 2020 news translation task. In Proceedings of the 5th Conference on Machine Translation (WMT), pages 145–155, 2020.

[4] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 355–362, 2011.

[5] Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. ParaCrawl: Web-scale acquisition of parallel corpora. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pages 4555–4567, 2020.

[6] Makoto Morishita, Jun Suzuki, and Masaaki Nagata. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC), pages 3603–3609, 2020.

[7] Wang Ling, Luís Marujo, Chris Dyer, Alan W. Black, and Isabel Trancoso. Crowdsourcing high-quality parallel data extraction from Twitter. In Proceedings of the 9th Workshop on Statistical Machine Translation (WMT), pages 426–436, 2014.

[8] Vassilis Papavassiliou, Prokopis Prokopidis, and Stelios Piperidis. Discovering parallel language resources for training MT engines. In Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC), 2018.

[9] Brian Thompson and Philipp Koehn. Vecalign: Improved sentence alignment in linear time and space. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1342–1348, 2019.

[10] Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. Transactions of the Association for Computational Linguistics (TACL), 7:597–610, 2019.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS), pages 6000–6010, 2017.

[12] Robert C. Moore and William Lewis. Intelligent selection of language model training data. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), pages 220–224, 2010.

[13] Roee Aharoni and Yoav Goldberg. Unsupervised domain clusters in pretrained language models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pages 7747–7763, 2020.

[14] Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. TICO-19: the translation initiative for COvid-19. In Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020, 2020.

[15] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pages 2818–2826, 2016.

[18] 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. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 48–53, 2019.

表 4　ファインチューニング時のハイパーパラメータ

Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$) [15]
学習率	1×10^{-5} (固定)
Dropout	0.3 [16]
Gradient Clipping	1.0
Label Smoothing	$\epsilon_{ls} = 0.1$ [17]
ミニバッチサイズ	16,000 トークン
アップデート回数	200 ステップ
Averaging	20 ステップごとにモデルを保存し、最終 8 モデルを平均し使用。
実装	`fairseq` [18]

A　ハイパーパラメータ

4 節で、既存汎用翻訳モデルを特定領域対訳コーパスを用いてファインチューニングする際のハイパーパラメータを表 4 に示す。