JParaCrawl: 大規模 Web ベース日英対訳コーパス

森下 睦 [1]，　鈴木 潤 [1,2]，　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

近年の機械翻訳手法は対訳コーパスから自動的に学習を行うため，このデータ量および品質が翻訳精度に大きな影響を与えることが知られている [24]．しかし，現在公開されている日英・日中などの日本語を含む対訳コーパスのデータ量は英仏間などと比べると小さく，また分野も限定的である．他の高資源言語対と同等の研究環境を整備するためには，大規模な日本語中心の対訳コーパスを作成する必要がある．本稿ではその最初の取り組みとして，Web から文章をクロールし大規模かつ幅広い分野を網羅した日英対訳コーパス “JParaCrawl” を作成することを考える．

また，現在のニューラル機械翻訳 (Neural Machine Translation, NMT) モデルの学習には大量の計算資源を必要とするため，用意できる計算機環境によっては実験が難しかった．我々は JParaCrawl で事前学習した NMT モデルを公開し，これを基に追加学習を行うことで特定分野向けの NMT モデルを少ない計算コストで作成できるようにする．

JParaCrawl および事前学習済みモデルは研究目的に限り Web から無料でダウンロード可能である[1]．なお本稿は arXiv に投稿した原稿 [11] を，最新の JParaCrawl (バージョン 2.0) で再実験し内容を一部修正したものである．

2 関連研究

対訳コーパスを作成する際のデータ源の一つに，国際機関の文書がある．これらから対訳データを作成した初期の成功例として，Europarl [7] がある．これは多言語に翻訳された欧州議会の議事録から，自動的に対訳文を抽出することで作成された大規模な対訳コーパスである．また，国連対訳コーパス [29] は同様に国連文書から作成された対訳コーパスである．これらの文書は専門家により翻訳されているため品質が高く，本文以外のメタデータも付与されていることが多いため対訳文抽出も容易であるが，対訳文の分野と言語対は限定的である．

もう一つの重要なデータ源は Web である．Smith らは大規模な Web クロールデータである Common Crawl[2] から大規模に対訳文を抽出する手法を提案した [25]．また，Schwenk らは Wikipedia から 1,620 言語対の対訳コーパスを作成した [22]．Web は幅広い分野と言語対を含んでおり，そのデータ量は日々増加している．そのため，データ源としては大きな可能性を秘めているが，高品質な対訳文を抽出するのは難しい場合が多い．

本研究は，近年の ParaCrawl プロジェクト[3] の成功に大きな影響を受けている．ParaCrawl プロジェクトは，Web をクロールし EU 公式言語-英語間の大規模な対訳コーパスを作成することを目標としている．同プロジェクトでは作成中の対訳コーパスを公開しており，すでに膨大な量となっている[4]．本コーパスは，すでに過去の WMT シェアードタスク [2, 1] でも使用されており，ParaCrawl は翻訳精度の向上に重要な役割を果たすことが参加者によって報告されている [5]．本研究では，ParaCrawl プロジェクトを発展させ，大規模な日英対訳コーパスを作成することを目標とする．

3 JParaCrawl

本コーパスは ParaCrawl プロジェクトと類似した手法で，Web から日英対訳文を収集した．本手法では，同一ドメイン上に日英対訳が存在すると思われる Web サイトを発見し，その Web データからの対訳文抽出を試みる．

クロール候補 Web サイトの選択　多くの日英対訳を含むドメインを発見するために，Common Crawl の全テキストデータに対して CLD2[5] により言語識別を行い，各ドメインの日英文量を得る．もし日英文量の比が同程度の場合，そのドメインは日英対訳文を含んでいる可能性が高い．そのため，この比をもとにクロール候補ドメインを約 15 万件選択した．

候補 Web サイトのクロール　その後，選択した候補 Web サイトをクロールする．Common Crawl データに含まれる Web データは，Web サイト全体を含んでいない，もしくはデータが古い可能性があるため，Web サイト全体を HTTrack[6] を用いて再度クロールした．本研究では，HTML ファイルに含まれるテキストデータのみに着目し，PDF など他のフォーマットについては収集していない．クロールはクラウドサービスを用いて行い，約 15 万ドメインをクロールした際のデータ量は gzip 圧縮後で 14.4TB となった．

対訳文抽出　次に，クロールした Web データから対訳文を抽出する．対訳文抽出には ParaCrawl

[1] http://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/
[2] https://commoncrawl.org/
[3] https://paracrawl.eu/
[4] 例として，独英対訳コーパスはすでに 3600 万文対を超えている．
[5] https://github.com/CLD2Owners/cld2
[6] http://www.httrack.com/

	文数	英単語数
フィルタリング前	39,703,172	913,083,258
フィルタリング後	10,120,013	238,636,621

表 1: JParaCrawl に含まれる文数および単語数

プロジェクトが公開している Bitextor[7] バージョン 7.0 を使用し，独自に日本語対応を行った．HTML ファイルからのテキスト抽出には日本語向けに開発された extractontent[8] を使用した．テキスト内の文分割には Moses [8] に付属している split-sentences.perl[9] を日本語に対応させ使用した．対訳文の抽出の際は，日本語文を内製の NMT モデルで英語に機械翻訳し BLEU スコア [17] が最も高くなる英文を対訳文として選択した [23].

表 1に収集した対訳コーパスの文数と単語数を示す．クロールデータからの対訳文抽出の結果，3970 万文を超える対訳文を得ることができた．しかし，この対訳コーパスは対訳アライメントの誤りなど大量のノイズを含んでしまっている．そのため，Bicleaner[10] [21] を用いてクリーニングを行う．フィルタリングモデルは内製の日英対訳コーパスを用い，スコアが 0.5 を下回ったものを削除した．また，同時に同一ドメイン内での重複文削除を行った．フィルタリング後，約 1012 万文のクリーンな対訳コーパスを得ることができた．

4 実験

本コーパスの有効性を確かめるために 2 つの実験を行った．4.1節では，JParaCrawl のみを用いて NMT モデルを学習し，様々な分野のテストセットで評価することでコーパスが幅広い分野を網羅していることを確認する．また，4.2節では，JParaCrawl を用いて事前学習した NMT モデルから特定の分野へ追加学習を行い，短時間て特定分野に特化した NMT モデルが学習可能であることを示す．以降の実験では，フィルタリング後の約 1012 万文を含む JParaCrawl を用いて実験を行う．

4.1 JParaCrawl を用いた NMT の学習

本節では，JParaCrawl を用いて NMT モデルを学習し複数のテストセットで評価を行うことで，JParaCrawl が幅広い分野を網羅していることを確認する．

4.1.1 実験設定

データ 本コーパスが幅広い分野を網羅していることを確かめるために，科学技術論文 (ASPEC [12]), 映画字幕 (JESC [20]), 京都に関連した Wikipedia 記事 (KFTT [13]), TED トーク (IWSLT 2017 の評価キャンペーンで用いられた tst2015 [3, 4]) の 4 つのテストセットを用いて評価を行った．コーパスは

[7] https://github.com/bitextor/bitextor

[8] https://github.com/yono/python-extractcontent

[9] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/ems/support/split-sentences.perl

[10] https://github.com/bitextor/bicleaner

データ数	文数	英単語数
ASPEC	3,008,500	68,929,413
JESC	2,797,388	19,339,040
KFTT	440,288	9,737,715
IWSLT	223,108	3,877,868

表 2: 学習データに含まれる対訳文数．ASPEC は本来 300 万文を含んでいるが，先行研究に基づきアライメントスコア上位 200 万文のみを学習に用いた [14].

	small	base	big
エンコーダ・デコーダ層数	6	6	6
Attention head 数	4	8	16
単語埋め込み層次元数	512	512	1,024
フィードフォワード層次元数	1,024	2,048	4,096

表 3: NMT モデル学習時のハイパーパラメータ

sentencepiece [9] を用いてサブワードに分割した．その際，語彙サイズは 32,000 に設定し，学習データについては 250 サブワード以上の文は削除した．なお，JParaCrawl が NFKC により正規化されているため，テストセットについても同様の正規化を行った．

比較のため，テストセットと同一分野の対訳コーパス (以下，インドメイン) のみを用いて NMT モデルの学習を行った．表 2に使用した学習データの文数および単語数を示す．ASPEC は対訳文アライメント時のスコアを基にデータがソートされており，先行研究に基づきスコアの上位 200 万文のみを学習に用いた [14].

NMT モデル 本実験では，fairseq [15] を用いて NMT モデルを学習した．NMT モデルは Transformer [27] を用い，small, base, big の 3 つのハイパーパラメータを用いて学習を行った．それぞれのハイパーパラメータの違いを表 3に示す．ASPEC および JESC での実験では big，KFTT では base，IWSLT では small のハイパーパラメータを使用した．すべての設定で，各層間のドロップアウト確率を 30%に設定した [26]. 最適化手法として，Adam [6] を $\alpha = 0.001$，$\beta_1 = 0.9$, $\beta_2 = 0.98$ とし使用した．学習率は Transformer の著者らに従い変化させ，ウォームアップステップは 4,000 に設定した [27]. 学習を安定化させるために，勾配のノルムが 1.0 を超えた場合それをクリップした [18]. ミニバッチサイズは 320,000 トークンに設定し，24,000 イテレーション学習を行った [16]. モデルは 200 イテレーション毎に保存し，デコード時には最後の 8 モデルを平均したものを使用した．また，GPU 上でより高速に学習を行うために混合精度演算を使用した [10]. デコード時はサイズ 6 のビームサーチを行い，その際各仮説のスコアを文長で割ることにより正規化を行った [28].

インドメインモデルについては，学習データサイズが小さいため IWSLT モデルのみミニバッチサイズを 80,000 トークンに変更した．また，開発用セットにより学習が収束したことを全てのインドメインモデルで確認したため，20,000 イテレーションで学習を打ち切った．

評価 翻訳精度の評価時には sacreBLEU [19] を用いて BLEU スコア [17] を計測した．sacreBLEU は日

本語を正しく単語分割できないため，英日翻訳結果については事前に MeCab[11] (IPA 辞書) を用いて単語分割を行った．

4.1.2 実験結果および考察

表 4に．インドメインデータのみ及び JParaCrawl のみで学習した NMT モデルの BLEU スコアを示す．JParaCrawl は幅広い分野を網羅することを目的としており特定分野に特化したコーパスではないため，インドメインデータのみを用いて学習した NMT をモデルより BLEU スコアは低くなっている．

しかし，これらのインドメインモデルは特定の分野のみに特化しているため，他の分野 (アウトオブドメイン) のデータについては翻訳精度が大幅に低下すると思われる．モデルの汎化性能を確認するため，学習分野以外の各テストセットから 1,000 文ずつ抽出し，3,000 文の分野外テストセットを作成した．表 5に分野外テストセットにおける BLEU スコアを示す．予期していた通り，インドメインデータから学習した NMT モデルは汎化性能に乏しく，分野外のテストセットについては翻訳精度が大幅に低下した．一方，JParaCrawl から学習したモデルは幅広い分野において安定した翻訳精度を維持できることを確認できた．

4.2 学習済みモデルからの追加学習

前節の実験結果から，JParaCrawl で学習した NMT をモデルは特定の分野には特化していないが，幅広い分野を網羅できていることが確認できた．このことから，本モデルを基に特定分野へ追加学習を行い，翻訳モデルをある分野に特化させることを考える．翻訳モデルを最初から学習するには大量の計算資源を必要とするが，事前学習モデルからの追加学習であれば低コストかつ短時間で特定分野の翻訳モデルを学習可能である．

4.2.1 実験設定

追加学習に用いたインドメインデータは4.1.1節で述べたものと同一である．4.1節で JParaCrawl を学習した際の最終保存モデルを事前学習モデルとして使用した．本モデルを初期値として，特定分野コーパスを用いて追加で 2,000 イテレーション学習した．その際のハイパーパラメータは4.1.1節で述べたものと同一である．なお，モデルの学習は 8 枚の NVIDIA RTX 2080 Ti GPU で行った．翻訳精度の評価は，前節と同様に BLEU スコアにより行う．

4.2.2 実験結果および考察

表 4に特定分野に対して追加学習を行った際の BLEU スコアを示す．JParaCrawl のみを用いて学習したモデルと比較すると，全ての設定で追加学習により大幅な精度向上が見られている．ASPEC および JESC を用いた実験ではインドメインデータで最初から学習を行ったモデルとほぼ同等の精度に達している．特に，ASPEC の英日翻訳ではインドメインモデルの精度をわずかに上回っている．また，KFTT および IWSLT を用いた実験では，英日・日英翻訳ともに大幅な精度向上が見られた．ASPEC や JESC はある程度のデータ量が存在しているのに対し，KFTT や IWSLT は学習データが小さいため JParaCrawl による事前学習の影響が大きかったと考えられる．

表 6にモデル学習に要した時間を示す．追加学習を用いた実験では，JParaCrawl を用いたモデルの事前学習にかかる時間は含めていない．モデルを初期から学習する場合と比較して，追加学習を用いて特定分野への適応にかかる時間は圧倒的に少ない．ゆえに，JParaCrawl で事前学習したモデルを追加学習することによって，初期から学習したモデルとほぼ同等もしくはそれ上回る NMT モデルを短時間かつ低コストに作成可能であることを示した．

なお，本実験で用いた事前学習済みモデルは研究目的に限り Web から無料でダウンロードできる[12]．

5 データセットの公開と著作権法

これまで本コーパスのように Web などから幅広くデータを収集しその成果物を公開する場合，成果物に他者の著作物が含まれているため著作権法上の問題が発生することが懸念されていたが，平成 31 年に施行された改正著作権法により著作物の利用制約が緩和された．本改正により，近年の情報通信技術の進展に対応したより柔軟な著作物の利用が認められ，特に改正後の第 30 条の 4 では，「広く著作物に表現された思想又は感情の享受を目的としない行為等を権利者の許諾なく行える」[13] ようになり，他者の著作物であってもコンピュータを用いた解析（翻訳モデル作成等）のためであれば広く利用することが可能になった．また，文化庁が公開している著作権改正に関する解説資料[14] によると，「自ら収集した学習用データを第三者に提供（譲渡や公衆送信等）する行為についても，当該学習用データの利用が人工知能の開発という目的に限定されていれば，本条に該当するものと考えられる」と述べられている．

6 おわりに

本稿では，大規模 Web ベース日英対訳コーパス JParaCrawl を紹介した．本コーパスは大規模に Web をクロールし，日英対訳文を自動的に収集し，ノイジーな対訳文対をフィルタリングすることで作成した．最終的に 1000 万文対以上の日英対訳データが得られ，これを無料で一般に公開した．本コーパスは現時点で一般に公開されている日英コーパスの中では最大である．

実験により，JParaCrawl が幅広い分野を網羅していることが確認でき，汎用的な翻訳モデル作成に有用であることを示した．また，JParaCrawl から学習したモデルをもとに特定分野のコーパスを用いて追加学習を

[11] https://taku910.github.io/mecab/

[12] http://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/

[13] http://www.bunka.go.jp/seisaku/chosakuken/hokaisei/h30_hokaisei/

[14] http://www.bunka.go.jp/seisaku/chosakuken/hokaisei/h30_hokaisei/pdf/r1406693_11.pdf

評価データ	英日翻訳			日英翻訳		
	インドメイン	**JParaCrawl**	追加学習	インドメイン	**JParaCrawl**	追加学習
ASPEC	44.3	26.5 (−17.8)	43.8 (−0.5)	28.7	19.7 (−9.0)	29.2 (+0.5)
JESC	14.5	6.5 (−8.0)	13.9 (−0.6)	17.8	7.5 (−10.3)	17.5 (−0.3)
KFTT	31.8	18.9 (−12.9)	33.2 (+1.4)	23.4	16.2 (−7.2)	25.9 (+2.5)
IWSLT	11.1	12.6 (+1.5)	14.5 (+3.4)	13.7	11.9 (−1.8)	17.2 (+3.5)

表 4: 学習した NMT モデルの BLEU スコア．括弧内の数値はインドメインモデルとの差を示す．インドメインおよび JParaCrawl 列の結果については4.1節を，追加学習列については4.2節を参照．

テスト除外データ	英日翻訳		日英翻訳	
	特定分野データのみ	**JParaCrawl**	特定分野データのみ	**JParaCrawl**
ASPEC	7.9	15.9 (+8.0)	5.7	15.4 (+9.7)
JESC	5.4	22.0 (+16.6)	8.6	18.2 (+9.6)
KFTT	4.6	18.3 (+13.7)	5.7	16.0 (+10.3)
IWSLT	5.0	19.6 (+14.6)	3.7	16.3 (+12.6)

表 5: 分野外テストセットでの BLEU スコア．括弧内の数値は特定分野データのみを用いて学習したモデルとの差を示す．

データ	インドメイン [h]	追加学習 [h]
ASPEC	13.22	1.60
JESC	14.19	1.69
KFTT	5.65	0.59
IWSLT	0.44	0.20

表 6: 英日 NMT モデルを学習するために必要な時間

行うことで，短時間で特定分野に適応できることを示した．

今後の課題として，より大規模な Web サイトのクロール，コーパス作成を行うことが挙げられる．さらに，対訳文アライメントやフィルタリングを高精度化することも重要な課題である．また，今後より多くの言語対に対応していくことも検討している．

謝辞 本研究にあたり，貴重なコメントを頂いた ParaCrawl プロジェクトに感謝する．また，技術的支援を頂いた NTT レゾナントの伊東久氏および浅井拓海氏に感謝する．

参考文献

[1] L. Barrault, O. Bojar, M. R. Costa-jussà, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, C. Monz, M. Müller, S. Pal, M. Post, and M. Zampieri. Findings of the 2019 conference on machine translation (WMT19). In Proc. WMT, pp. 1–61, 2019.

[2] O. Bojar, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, and C. Monz. Findings of the 2018 conference on machine translation (WMT18). In Proc. WMT, pp. 272–303, 2018.

[3] M. Cettolo, M. Federico, L. Bentivogli, J. Niehues, S. Stüker, K. Sudoh, K. Yoshino, and C. Federmann. Overview of the IWSLT 2017 evaluation campaign. In Proc. IWSLT, pp. 2–14, 2017.

[4] M. Cettolo, C. Girardi, and M. Federico. WIT3: web inventory of transcribed and translated talks. In Proc. EAMT, pp. 261–268, 2012.

[5] M. Junczys-Dowmunt. Microsoft's submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In Proc. WMT, pp. 425–430, 2018.

[6] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In Proc. ICLR, 2015.

[7] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In Proc. MT Summit, pp. 79–86, 2005.

[8] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In Proc. ACL, pp. 177–180, 2007.

[9] T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proc. EMNLP, pp. 66–71, 2018.

[10] P. Micikevicius, S. Narang, J. Alben, G. F. Diamos, E. Elsen, D. García, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. Mixed precision training. In Proc. ICLR, 2018.

[11] M. Morishita, J. Suzuki, and M. Nagata. JParaCrawl: A large scale web-based japanese-english parallel corpus. arXiv preprint arXiv:1911.10668, 2019.

[12] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. ASPEC: Asian scientific paper excerpt corpus. In Proc. LREC, 2016.

[13] G. Neubig. The Kyoto free translation task. `http://www.phontron.com/kftt`, 2011.

[14] G. Neubig. Forest-to-string SMT for asian language translation: NAIST at WAT2014. In Proc. WAT, pp. 20–25, 2014.

[15] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modeling. In Proc. NAACL, pp. 48–53, 2019.

[16] M. Ott, S. Edunov, D. Grangier, and M. Auli. Scaling neural machine translation. In Proc. WMT, pp. 1–9, 2018.

[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In Proc. ACL, pp. 311–318, 2002.

[18] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In Proc. ICML, pp. 1310–1318, 2013.

[19] M. Post. A call for clarity in reporting BLEU scores. In Proc. WMT, pp. 186–191, 2018.

[20] R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz. JESC: Japanese-English Subtitle Corpus. arXiv preprint arXiv:1710.10639, 2017.

[21] V. M. Sánchez-Cartagena, M. Bañón, S. Ortiz-Rojas, and G. Ramírez-Sánchez. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In Proc. WMT, pp. 955–962, 2018.

[22] H. Schwenk, V. Chaudhary, S. Sun, H. Gong, and F. Guzmán. WikiMatrix: Mining 135m parallel sentences in 1620 language pairs from Wikipedia. arXiv preprint arXiv:1907.05791, 2019.

[23] R. Sennrich and M. Volk. Iterative, MT-based sentence alignment of parallel texts. In Proc. NODALIDA, pp. 175–182, 2011.

[24] R. Sennrich and B. Zhang. Revisiting low-resource neural machine translation: A case study. In Proc. ACL, pp. 211–221, 2019.

[25] J. R. Smith, H. Saint-Amand, M. Plamada, P. Koehn, C. Callison-Burch, and A. Lopez. Dirt cheap web-scale parallel text from the common crawl. In Proc. ACL, pp. 1374–1383, 2013.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15:1929–1958, 2014.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In Proc. NIPS, pp. 6000–6010, 2017.

[28] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.

[29] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen. The united nations parallel corpus v1.0. In Proc. LREC, pp. 3530–3534, 2016.