多言語モデルを用いた日英対訳文集合のフィルタリング手法の分析

酒井大樹 [1]　宇津呂武仁 [1]　永田昌明 [2]
[1] 筑波大学大学院 システム情報工学研究群 [2]NTT コミュニケーション科学基礎研究所
s2220746_@u.tsukuba.ac.jp　utsuro_@iit.tsukuba.ac.jp　masaaki.nagata_@ntt.com

概要

本論文では，ウェブクロールコーパスである JParaCrawl-v3 に含まれる不適切な文対を，多言語モデルによるスコアを用いてフィルタリングすることを目指した．事前に異なる文の断片を含む文対応誤りデータを擬似的に作成し，その誤りの検出する様子を 4 つの言語モデルの間で調査した．その結果，Bicleaner-AI と marian-scorer の方が LEALLA，GPT-2 よりも正しく誤りを検出する度合いが大きかった．その後，各モデルを用いて JParaCrawl-v3 のフィルタリングを実行した結果，Bicleaner-AI と marian-scorer は，フィルタリングにおいても英日方向に有意差を出すことがわかった．

1　はじめに

ニューラルモデルによる機械翻訳において，モデルの訓練に使用されるデータの品質はモデルが出力する翻訳文の品質に大きな影響を与える．しかしながら，訓練に使用する対訳文集合はウェブクロールによって収集された対訳文集合を使用する場合が多く，これは自動収集という性質上，ノイズとなるデータを一定割合含むために，事前に処理を行うことでその品質を向上させることが必要である．この問題を解決することを目的として，WMT2018 から WMT2020 までの Parallel Corpus Filtering Task，WMT2023 の Parallel Data Curation Task が開催されるなど，対訳文集合の品質保証は課題となっている．対訳文集合のフィルタリングにおいて，最も用いられるスコア付け手法の一つは，多言語文埋め込みモデル LASER [1] であり，これは，WMT2020，2023 のベースラインとしても使用されている．

本論文では，WAT2022 で実施された Parallel Corpus Filtering Task [1)] の条件に沿って，日英間で最大規模の対訳文集合である JParaCrawl-v3 をフィルタリングすることを目指す．これは多様なドメインデータを含む対訳文集合であるので，ドメイン非依存のモデルを用いて JParaCrawl-v3 をフィルタリングすることを最終目的とする．まず文対応に関する擬似的な誤りデータを作成し，それに対する反応の様子から Bicleaner-AI，marian-scorer，LEALLA，GPT-2 スコアの性能を評価した．その結果，Bicleaner-AI，marian-scorer の 2 手法が正しく誤りを検出する度合いが大きかった．その後，前処理を行った JParaCrawl-v3 の各文に 4 種スコアを与え，それに基づくフィルタリングを行い，フィルタリング後対訳文集合を獲得した．これを用いて評価用機械翻訳モデルを訓練し，テストデータに対する BLEU 値をもとにフィルタリング手法を評価すると，日英方向において Bicleaner-AI と marian-scorer は有意差を示した．4 種スコアを組み合わせた合成スコアに関しても，類似した傾向を示した．

2　JParaCrawl-v3: ウェブから収集された日英対訳文対集合

Morishita ら [2] は，CommonCrawl を分析して収集した対訳 Web サイト候補と，クラウドソーシングを使って収集した対訳 Web サイト候補から JParaCrawl-v3 を構築した．この文集合は次の 3 段階に分けて構築され，対訳文集合として公開されている．

1. CommonCrawl 内のデータのうち，CLD [2)] を用いて日本語，英語が同程度含まれるドメインを特定し，対訳文抽出の候補サイトとする．
2. HTTrack [3)] を用いてさらにウェブサイトを収集し，候補サイトに追加した．
3. Bitextor toolkit [4)] を用いて，対訳文対を抽出する．

1） https://sites.google.com/view/wat-filtering/previous-task-wat-2022?authuser=0

2） https://github.com/CLD2Owners/cld2
3） http://www.httrack.com/
4） https://github.com/bitextor/bitextor

本論文では，上記の方法で構成された，2,500 万文からなる JParaCrawl-v3 に対し，複数の前処理を施し 1,600 万文まで削減した JParaCrawl-v3 を用いる．前処理の詳細は付録に記載する．

3 言語モデルを用いたスコア付け

3.1 Bicleaner-AI

Bicleaner-AI [5] は公開されている日英翻訳モデルであり，原言語文 x から目的言語文 y に翻訳される確率を与えるモデルである．この確率をもとに日英文対にスコアを与える．

3.2 marian-scorer

本論文では Marian [3] を用いてニューラル翻訳モデル [6] を構築し，JParaCrawl-v3 のデータを用いてモデルの訓練を行った．訓練データとして JParaCrawl-v3 より無作為に抽出した日英 100 万文対を利用する．Junczys-Dowmunt [4] の方法に従い，日英と英日の双方向で Transformer ベースの翻訳モデルを訓練し，このモデルを用いて文対のスコア付けを行った．具体的には，言語対 (x, y) があるとき，次の式により dual conditional cross-entropy を計算する．

$$|H_A(y|x) - H_B(x|y)| \quad + \frac{1}{2}(H_A(y|x) + H_B(x|y))$$

ただし，A，B はそれぞれ日英とその逆方向の翻訳モデルであることを示している．$H_M(\cdot|\cdot)$ はモデル M に関する確率分布 $P_M(\cdot|\cdot)$ による正規化済みの単語の conditional cross-entropy であり，次の式のように表される．x, y が逆の場合も同様である．

$$\begin{aligned} H_M(y|x) &= -\frac{1}{|y|}\log P_M(y|x) \\ &= -\frac{1}{|y|}\sum_{t=1}^{|y|}\log P_M(y_t|y_{<t}, x) \end{aligned}$$

このスコアは，cross-entropy の絶対値の差を調べる部分 $|H_A(y|x) - H_B(x|y)|$ と，低い cross-entropy に対するペナルティ部分 $\frac{1}{2}(H_A(y|x) + H_B(x|y))$ の 2 つからなる．さらに以下の式変形を施してスコア adq*(x,y)* を得る．これにより値域が [0, 1] に収まる．

$$\begin{aligned} \text{adq}(x,y) = \exp\big(&-(|H_A(y|x) - H_B(x|y)| \\ &+\frac{1}{2}(H_A(y|x) + H_B(x|y)))\big) \end{aligned}$$

5) https://huggingface.co/bitextor/bicleaner-ai-full-en-ja

6) https://marian-nmt.github.io/docs/cmd/marian/

3.3 LEALLA

本論文では，最大 2,000 万文対に対してスコア付けを行うにあたり，より短時間で実験を行うために多言語文埋め込みモデル LaBSE [1] の軽量実装版である setu4993/LEALLA-large [7] を用いて文埋め込みベクトルを取得した．このベクトルを $L2$ 正規化して余弦類似度を計算し、対訳スコアとする．

3.4 GPT-2

GPT-2 による文スコアとして，言語モデルから得られる Perplexity を用いた．日本語モデルとしては rinna/japanese-gpt-1b [8]，英語モデルとして gpt2-large [9] を用いた．

3.5 合成スコア

原言語文 x，目的言語文 y がある時，3.1 節，3.2 節，3.3 節，3.4 節のモデルを通して得られるスコアの和を用いる．すなわち，モデル x のスコアを $score(x)$，モデル y のスコアを $score(y)$ としたとき，その合成スコアを $S = score(x) + score(y)$ と計算し、S をもとにフィルタリングする．ただし，GPT-2 のスコアは値域が異なるため，標準化を行い，平均値 0，分散 1 としてある．

4 誤訳を混入した擬似データに対する言語モデルの特性の分析

本節では，誤訳のうち文の対応が誤っている例を想定して擬似的に誤訳データを作成する方法を示し，それらに対する各言語モデルの反応を分析する．正しい対訳では文 x_1，文 x_2 が翻訳として対応していることが期待されるが，自動的に対訳データを収集する際，x_1，x_2 のいずれかの前方または後方に無関係な文の一部が混入してしまうことがある．この誤りを本論文では文対応誤りと呼び，これを持つ擬似データを作成する．

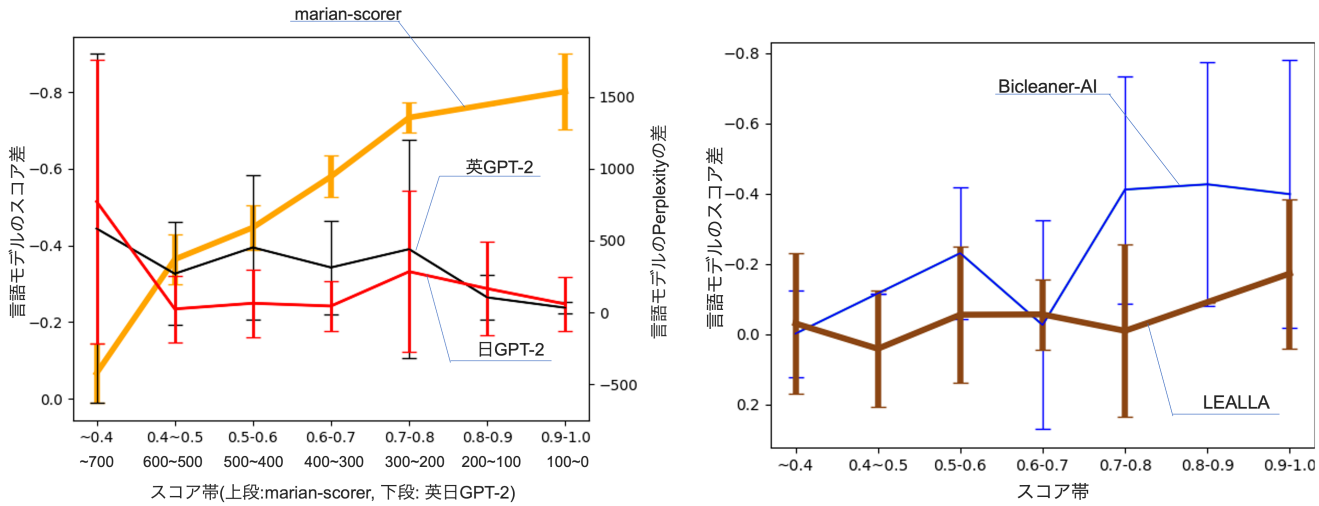
4.1 擬似データ作成手順

翻訳として対応する言語 A，言語 B の文対 $x = (x_A, x_B)$，$y = (y_A, y_B)$ を定義する．続いて，文 y_A，文 y_B から誤り混入用断片 $yhead_A$，$ytail_A$，$yhead_B$，$ytail_B$ を作成する．この時，誤り混入用断片は全て

7) https://huggingface.co/setu4993/LEALLA-large

8) https://huggingface.co/rinna/japanese-gpt-1b

9) https://huggingface.co/gpt2-large



(a) 言語モデルのスコア (marian-scorer) 差・Perplexity のスコア (GPT-2) 差の分布

(b) 言語モデルのスコア (Bicleaner-AI，LEALLA) 差の分布

図 1　言語モデルのスコア (GPT2 以外) 差・Perplexity のスコア (GPT-2) 差の分布

表 1　各言語モデルによるフィルタリング性能の比較 (すべて前処理適用済みの 1,600 万文に対するフィルタリング．*は，無作為抽出 100 万文で訓練したベースライン翻訳モデルとの間で BLEU 値の有意差 ($p < 0.05$) があることを示す．)

フィルタリング手法	英日		日英	
テストデータの種類	ASPEC	WMT22	ASPEC	WMT22
無作為抽出 100 万文	13.8	17.1	16.7	14.8
Bicleaner-AI 上位 100 万文	14.6*	17.9*	16.2	15.1
marian-scorer 上位 100 万文	15.5*	18.4*	16.5	14.8
LEALLA 上位 100 万文	13.7	16.3	17.4	14.5
GPT-2 上位 100 万文	12.0	14.2	12.0	9.7
Bicleaner-AI + GPT-2 上位 100 万文	14.4	16.9	15.6	14.3
LEALLA + marian-scorer 上位 100 万文	15.4*	18.1*	17.2	15.2

文字長 10 で固定されていて，文 y_A，文 y_B の文頭および末尾から切り取ったものである．この時，文対応誤りを含む擬似データの作成手順は以下である．

1. 文 x_A の文頭に $ytail_A$ を付加し文 $x\_headerror_A$ を作成．同様に文 x_B の文頭に $ytail_B$ を付加し文 $x\_headerror_B$ を作成する．
2. 文 x_A の末尾に $yhead_A$ を付加し文 $x\_tailerror_A$ を作成，同様に文 x_B の末尾に $yhead_B$ を付加し文 $x\_tailerror_B$ を作成する．

以上より，誤りを含む文対

$$x\_headerror = (x\_headerror_A, x\_headerror_B)$$
$$x\_tailerror = (x\_tailerror_A, x\_tailerror_B)$$

の 2 例を作成した．本論文では，JParaCrawl-v3 から無作為抽出し人手で適切と判定した文対を使用する．x を 100 文対，y を 100 文対用意し，合計 20,000 文対の擬似データを作成した．

4.2　言語モデルの適用手順および分析結果

3 節で示した 4 つの言語モデルが，4.1 節で作成された擬似データに対し与えるスコアを調べた．誤りを作成する元となった文対 $x = (x_A, x_B)$ に対し，擬似データは 200 文対作成される．この 200 文対のスコアが文対 x のスコアに対しどれほど変化したのかを調べ，結果を図 1 に記載した．

これによると Bicleaner-AI，marian-scorer は，元の文のスコアが高い場合の文対応誤りに対して反応し，スコアが下落している様子が見られる．一方で，GPT-2 では，元の文の Perplexity が高い場合に文対応誤りに対して反応し，Perplexity が上昇している様子が見られる．また，GPT-2 の日本語，英語を比較すると，全体的に英語の方が Perplexity の上昇幅が大きく，英語の方が日本語より誤りへの反応がやや鋭いと言える．LEALLA はいずれの傾向も強く現れることがなかった．スコアが低い文対は元々

フィルタリングによる除去対象であるため，スコアが高い場合に対して反応が見られる Bicleaner-AI，marian-scorer の 2 手法は，文対応誤りの観点から適切なフィルタリングができていると言える．

5 フィルタリング評価

5.1 評価方法

JParaCrawl-v3 の各文に各言語モデルのスコア付けを行い，上位 100 万文を取り出すことでフィルタリング済み対訳文集合を獲得する．このフィルタリングの評価のために，同集合を用いて評価用翻訳モデルを訓練し，テストデータに対する BLEU 値で評価する．また，ベースライン翻訳モデルとして，無作為抽出した 100 万文で訓練した翻訳モデルによる機械翻訳文との間で翻訳精度の有意差検定を行なった．この検定においては mteval ツール [10] を用いブートストラップ法によって行った．

評価のため，Fairseq ライブラリを用いて Transformer ベースのニューラル機械翻訳モデルを作成した．パラメータは WAT2022 に従うが，訓練時間の都合上，一部調整している．詳細は付録に記載する．フィルタリング済みの 100 万対訳文対を用いてこのモデルの訓練を行い，ASPEC [5] テストデータ，WMT22 テストデータを対象として，BLEU 値を算出して日英・英日翻訳の翻訳精度の評価を行った．

5.2 評価結果

表 1 より，全体としてフィルタリングにより有意差が現れるのが英日方向の場合に限定されているのは，JParaCrawl-v3 に元々含まれる日本語文の品質が低く，無作為抽出した文集合を用いて訓練した翻訳モデルが生成する日本語文の品質が低いためだと考えられる．WMT-2023 の official results における SKIM チームの日英翻訳 (2 位) と英日翻訳 (5 位) の順位の差も JParaCrawl-v3 から作成した翻訳モデルは日本語を生成する能力が低いことを示唆している [6]．従って，英日翻訳では，本論文で有意差があると示された Bicleaner-AI や marian-scorer を用いて JParaCrawl-v3 をフィルタリングすることによって，翻訳精度が改善する可能性がある．

また，英日方向で単独でも有意差が現れるのは Bicleaner-AI，marian-scorer であった．合成スコアに関しては，有意差が現れる場合と現れない場合があり，明確な結論を述べることができない．今後 4 種類のスコアの内の 2 種類の組み合わせ全 6 通りの内，残りの Bicleaner-AI と marian-scorer，Bicleaner-AI と LEALLA，LEALLA と GPT-2，marian-scorer と GPT-2 の組を実験する必要がある．

10） https://github.com/odashi/mteval

6 関連研究

Bane ら [7] は，Wikipedia から収集した CCMatrix データセットを対象として，対訳文のエラーを 10 種類に分類することで各種フィルタリングモデルによるフィルタリング効果を調査した．この調査では，marian-scorer[3] ツールを用いた Dual conditional cross-entropy[4] による手法が最も優れていると指摘されている．一方で，この手法は訳抜けの誤訳に対応することができず，そのような誤訳を許容してしまうフィルタリングをするとも指摘されている．同時に，多言語モデルによるフィルタリングでは訳抜けが防がれているという指摘もなされた．本論文では，これらフィルタリングモデルを組み合わせることの有意性を示しており，日英言語対において各手法の欠点を相補できたといえる．

また，WAT2022 の Parallel Corpus Filtering Task では，Feature Decay Algorithm が日英間でベスト [8] であったが，本論文ではドメイン非依存手法を用いる目的があり，同論文の手法は比較しなかった．

7 おわりに

本論文では JParaCrawl-v3 に対して，翻訳モデル，言語モデルを用いることで品質の悪い文対をフィルタリングすることを提案した．擬似誤り分析によって良い傾向が発見された手法は，後のフィルタリングでも，無作為抽出した場合と比べて BLEU 値による評価で有意差が観測された．また，擬似誤り分析で良い傾向であった手法同士の合成スコアによるフィルタリングでも有意差が観測された．今後は，擬似誤りデータによる分析範囲を数値誤り，固有名詞誤りなどに拡大して，フィルタリング結果との相関を調べていくことになると思われる．

参考文献

[1] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. Language-agnostic BERT sentence embedding. In **Proc. 60th ACL**, pp. 878–891, 2022.

[2] M. Morishita, J. Suzuki, and M. Nagata. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In **Proc. 12th LREC**, pp. 3603–3609, 2020.

[3] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. Aji, N. Bogoychev, A. F. T. Martins, and A. Birch. Marian: Fast neural machine translation in C++. In **Proc. 56th ACL, System Demonstrations**, pp. 116–121, 2018.

[4] M. Junczys-Dowmunt. Dual conditional cross-entropy filtering of noisy parallel corpora. In **Proc. 3rd WMT**, pp. 888–895, 2018.

[5] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. ASPEC: Asian scientific paper excerpt corpus. In **Proc. 10th LREC**, pp. 2204–2208, 2016.

[6] Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In Philipp Koehn, Barry Haddow, Tom Kocmi, and Christof Monz, editors, **Proceedings of the Eighth Conference on Machine Translation**, pp. 1–42, Singapore, December 2023. Association for Computational Linguistics.

[7] F. Bane, C. S. Uguet, W. Stribiżew, and A. Zaretskaya. A comparison of data filtering methods for neural machine translation. In **Proc. 15th AMTA (Vol. 2: Users and Providers Track and Government Track)**, pp. 313–325, 2022.

[8] A. Poncelas, J. Effendi, O. Htun, S. Yadav, D. Wang, and S. Jain. Rakuten's participation in WAT 2022: Parallel dataset filtering by leveraging vocabulary heterogeneity. In **Proc. 9th WAT**, pp. 68–72, Gyeongju, Republic of Korea, 2022.

A　前処理

本付録では，JParaCrawl-v3 に対し，取り除く理由が明らかな文対をフィルタリングする方法について述べる．JParaCrawl-v3 を母集団 A とした時，文対 $n(n \in A)$ は全て日本語文 ja，英語文 en の日英文対からなり，次のように表現される．

$$n \in A, n = (en, ja)$$

A.1　言語 ID

言語 ID とは，fasttext [11] の言語判定モデル [12] を用いて推定される文の言語の種類によるフィルタリングを言う．JParaCrawl-v3 は英語文，日本語文の順序で配置されているので，この構造を利用して英日それぞれの言語 ID が合っていない文対を取り除く．

A.2　トークン数の上限

語彙サイズを 32,000，JParaCrawl-v3 全体を学習データとして訓練を行った sentencepiece [13] により，日本語文、英語文のトークン分割を行う．この時、集合 A の各要素 n に対し，各文を sentencepiece でトークン分割した時，en，ja いずれかに含まれるトークン数が 150 以上である文対を取り除く．

A.3　数字の不一致

ある文対 $n = (en, ja)$ それぞれからアラビア数字を抽出し，ソートしたものを比較し，それが一致する文対のみを採用する．一致条件として，Python プログラム上で同じ数字として扱われることで判定をする．

A.4　重複文対

集合 A の各要素 n に対し，$n_1 = (en_1, ja_1)$, $n_2 = (en_2, ja_2)(n_1, n_2 \in A)$ なる文対を 2 組取り出した時，$ja_1 = ja_2$ となるような文対であれば n_2 を取り除く．

B　擬似データの具体例

誤りを含まないベース文対として，

英語	The cones are covered with thyroid flakesthat hide plant seeds.
日本語	コーンは植物の種を隠す甲状腺フレークで覆われています。

がある時，誤り断片を追加した文対応誤り文は以下が作成される．まずベース文対の前方に誤りが付加された $x\_headerror_B$ の例として，

英語	apid pace. The cones are covered with thyroid flakes that hide plant seeds.
日本語	高を増加させました。コーンは植物の種を隠す甲状腺フレークで覆われています。

次に，ベース文対の後方に誤りが付加された $x\_tailerror_B$ の例として，

英語	The cones are covered with thyroid flakes that hide plant seeds. Foreign re
日本語	コーンは植物の種を隠す甲状腺フレークで覆われています。その上で、速いペース

この様に作成され，1 つのベース文対から 2 種類の文対応誤りを含む日英文対を作成される．

C　評価用モデルのパラメータ

基本的には WAT2022 の設定 [14] に従うが，warm-up updates，max-update を変更している

表 2　評価用モデルのパラメータ

パラメータ名	値
warmup-updates	800
max-update	10000

11）https://github.com/facebookresearch/fastText
12）https://fasttext.cc/docs/en/language-identification.html
13）https://github.com/google/sentencepiece
14）https://github.com/MorinoseiMorizo/wat2022-filtering/blob/main/train_model_big_enja.sh