2024 年度

修　士　論　文

BART ファインチューニングを用いた機械翻訳の事前編集

指導教員: 村上陽平

立命館大学大学院 情報理工学研究科
情報理工学専攻 博士課程前期課程
計算機科学コース

学生証番号: 6611220075-5

氏名: ZHANG Yuxuan

BART ファインチューニングを用いた機械翻訳の事前編集

Zhang Yuxuan

内容梗概

ニューラル機械翻訳の登場により,機械翻訳がより高い翻訳精度を達成している．その結果,ニューラル機械翻訳は,多言語コミュニケーションにおいてますます重要な役割を果たしている．特に,国際交流の場では,Lingua Franca と呼ばれ世界の共通言語として用いられる英語でのコミュニケーションが必要とされており,その負担を軽減するために,非母語話者は機械翻訳を多用する傾向にある.

しかしながら,ニューラル機械翻訳が必ずしも英語母語話者の英語を生成するとは限らないため,コミュニケーションの齟齬を生じさせる可能性がある．これは,大規模な機械翻訳モデルが,ウェブ上の大量の対訳ペアを収集してモデルを構築するのが一般的であり,英語非母語話者の英語も機械翻訳の学習対象に含められるためである.

そこで,日本語から英語への翻訳において,英語母語話者の英語を生成するための機械翻訳の事前編集手法を提案する.事前編集とは,機械翻訳で翻訳する前に原文を機械翻訳に適した形に変換することである．特に,本研究では,現状の機械翻訳を通して英語母語話者の英語を生成しやすい日本語文を「翻訳しやすい日本語」と呼び,原文を翻訳しやすい日本語に変換することを目的とする.具体的には,日英の対訳コーパスから日本語文と翻訳しやすい日本語のペアを抽出し,このデータで BART をファインチューニングすることで,日本語文から翻訳しやすい日本語への変換器を構築する．本手法の実現にあたり,取り組むべき課題は以下の 2 点である.

日本語文–翻訳しやすい日本語文のペアの取得

日本語文から翻訳しやすい日本語への日本語変換器を構築するために,日本語文と翻訳しやすい日本語文のペアが必要である．日本語文は変換前の翻訳しにくい日本語でなければならない．翻訳しにくい日本語は機械翻訳によって日本人英語が生成される日本語文であるため,日本語文を一度機械翻訳によって翻訳しなければならず,大規模なデータを構築するには,翻訳コストが非常に高い．そこで,翻訳コストを抑えるために,できる限り翻訳しにくい日本語と推定されるものに原文段階で限定しなければならない.

日本語文から翻訳しやすい日本語への日本語変換器

日本語文と翻訳しやすい日本語文のペアを用いて日本語変換器を構築する必要がある．しかしながら,変換モデルを学習するには,日本語文と翻訳しやすい日本語文のペアが大規模に必要となるため,事前学習済みのモデルを用いなければならない.

一つ目の課題に対しては,母語話者英語を機械翻訳で訳したら翻訳しやすい日本語を得られるため,日本語文と母語話者英語文のペアを収集する必要がある.本研究では方法は二つある.一つ目の方法は Wikipeadia の日本語版と英語版の中に同じ意味を持つ文を日本語と英語のペアとして抽出する．二つ目の方法は英語の分類器を用い,日英の対訳コーパスから日本語と母語話者英語のペアを抽出する.具体的には,日英ペアの英語部分は母語話者英語のペアをまず抽出する.次に,翻訳コストを節約するために分類モデルを用い,翻訳しやすい日本語を含む文ペアをさらにフィルタリングする．具体的には,実際に翻訳しやすい日本語文と翻訳しにくい日本語文を収集し,それらを用いて,日本語文の翻訳難易度を判定する分類器を学習した.その結果,翻訳しにくい日本語文に対して精度 0.6,再現率 0.3 の分類器ができた.次に,日本語と母語話者英語のペアを機械翻訳で日本語文と翻訳しやすい日本語文のペアを収集する.

二つ目の課題に対しては,事前訓練された日本語版の BART モデルを用いる.BART とは,Transformer を大量のテキストデータで事前学習したモデルである.BART をファイチュニングし,日本語文から翻訳しやすい日本語への日本語変換器を構築する．変換前後の日本語文の英訳の母語話者英語レベル判定を用いて評価を行い,提案手法の有効性を検証する．本研究の貢献は以下の通りである.

日本語文と母語話者英語のペアの取得

二つの手法で日本語と母語話者英語のペアを取得した．Wikipeadia の日本語版と英語版の中から約 700 件のペアを作った．さらに,Jpara という日英コーパスの 770 万のペアをフィルタリングし,約 13,000 件の日本語文と母語話者英語のペアを取得した.翻訳が難解な日本語文に対して精度 0.6,再現率 0.3 の分類器を得ることができた.

日本語文から翻訳しやすい日本語への変換器の構築

日本語バージョンの BART モデルをベースにファイチュニングし,日本語文から翻訳しやすい日本語への日本語変換器を構築した．変換前後の日本語文の英訳結果の母語化程度を用いて評価を行い,変換器が異なるデータセットに対する改善性能は 2%~22%である.

Pre-editing machine translation using BART fine tuning

Zhang Yuxuan

Abstract

With the advent of neural machine translation, machine translation has achieved higher translation accuracy. As a result, neural machine translation is playing an increasingly important role in multilingual communication. Especially in the context of international exchange, communication in English, which is referred to as the Lingua Franca and used as the common language of the world, is required. To alleviate this burden, non-native speakers tend to use machine translation extensively.

However, neural machine translation does not necessarily generate English that is native-like, which can lead to communication discrepancies. This is because large-scale machine translation models are typically built by collecting a vast number of parallel text pairs from the web, which includes English written by non-native speakers as well.

Therefore, this study proposes a pre-editing method for machine translation to generate English that is native-like in translations from Japanese to English. Pre-editing involves converting the original text into a form suitable for machine translation before the actual translation process. Specifically, this research defines "easy to translate Japanese" as Japanese sentences that are more likely to generate native-like English through the current machine translation system. The goal is to convert the original text into "easy to translate Japanese."In detail, pairs of original Japanese sentences and "easy to translate Japanese" will be extracted from a Japanese-English parallel corpus. Using this data, BART will be fine-tuned to create a converter that transforms original Japanese into "easy to translate Japanese." The two main challenges to address in implementing this method are as follows:

Obtaining Pairs of Original Japanese and Easy-to-Translate Japanese Sentences

To construct a converter that transforms original Japanese sentences into easy-to-translate Japanese, it is necessary to have pairs of original Japanese sentences and easy-to-translate Japanese sentences. The original Japanese sentences must be those that are difficult to translate. Difficult-to-translate Japanese sentences are those that generate non-native English through machine translation, so the original Japanese sentences must first be translated via machine translation. This makes the translation cost very high when constructing large-scale data. Therefore, to minimize translation

costs, it is essential to restrict the original sentences as much as possible to those estimated to be difficult to translate at the initial stage.

Japanese Converter for Transforming Japanese into Easy-to-Translate Japanese

It is necessary to construct a Japanese converter using pairs of original Japanese sentences and easy-to-translate Japanese sentences. However, to train the conversion model, a large-scale dataset of these pairs is required, necessitating the use of a pre-trained model.

To address the first challenge, it is necessary to collect pairs of original Japanese sentences and native English sentences. This study proposes two methods: the first is to extract pairs of sentences with the same meaning from the Japanese and English versions of Wikipedia; the second is to use an English classifier to extract pairs from a Japanese-English parallel corpus. Specifically, the English part is screened to identify native English sentences. To save translation costs, a classification model is used to further filter pairs that include easy-to-translate Japanese. A classifier is trained to determine the translation difficulty of Japanese sentences, achieving an accuracy of 0.6 and a recall of 0.3 for difficult-to-translate sentences.

For the second challenge, a pre-trained Japanese BART model is used. BART, a Transformer model pre-trained on a large amount of text data, is fine-tuned to construct a converter from original Japanese to easy-to-translate Japanese. The effectiveness of the proposed method is verified using native speaker-level English evaluations of translations before and after conversion.The contributions of this study are as follows:

Acquisition of Japanese and Native English Sentence Pairs:

Two methods were used to acquire pairs of Japanese and native English sentences. About 700 pairs were created from the Japanese and English versions of Wikipedia. Additionally, by filtering 7.7 million pairs from the Jpara Japanese-English corpus, approximately 13,000 pairs of Japanese and native English sentences were obtained.

Construction of a Converter from Original Japanese to Easy-to-Translate Japanese:

A converter was constructed by fine-tuning a Japanese version of the BART model to transform original Japanese into easy-to-translate Japanese. Evaluations were conducted using the native-likeness of the English translation results before and after conversion, showing an improvement in the native speaker level of machine translations by approximately 2% to 22%.

目次

第1章 はじめに

ニューラル機械翻訳の登場により,機械翻訳がより高い翻訳精度を達成している. その結果,ニューラル機械翻訳は,多言語コミュニケーションにおいてますます重要な役割を果たしている. 特に,国際交流の場では,Lingua Franca と呼ばれ世界の共通言語として用いられる英語でのコミュニケーションが必要とされており,その負担を軽減するために,非母語話者は機械翻訳を多用する傾向にある.

翻訳モデルの訓練には,大量のテキストデータが必要である. 通常,ネット上で収集した大量のコーパスを訓練データとする. これらのコーパスには,英語母語話者の英語だけではない,世界中の英語学習者の英語が含まれている.同じ英語文とはいえ,英語文には母語や文化圏などの影響で人々のバイアスが混ざっており,翻訳モデルがこれらのバイアスを学習し,翻訳結果に反映される.その結果,機械翻訳の英語文には,英語母語話者の英語だけではない,英語非母語話者の英語も含まれる.例えば,「経験値に差がありすぎて,そうした傾向になりがちです。」という文を Deepl で英語に翻訳したら「Too much difference in experience tends to make such a tendency.」という英語文になる.英語母語話者か日本語話者かを判別できる英語スタイル分類器により,この英語は英語母語話者の英語ではない. 分類器により,母語英語話者の英語は「There is too much difference in experience, and this tends to be the case」だが今の機械翻訳は生成できない.そして,ニューラル機械翻訳が必ずしも英語母語話者の英語を生成するとは限らないため,書き手の本来の意図とは異なった意図が伝達され,コミュニケーションの齟齬を生じさせる可能性がある.

そこで,日本語から英語への翻訳において,英語母語話者の英語を生成するための機械翻訳の事前編集手法を提案する. 事前編集とは,機械翻訳で翻訳する前に原文を機械翻訳に適した形に変換することである. 特に,本研究では,現状の機械翻訳を通して英語母語話者の英語を生成しやすい日本語文を「翻訳しやすい日本語」と呼び,原文を翻訳しやすい日本語に変換することを目的とする. 具体的には,日英の対訳コーパスから日本語文と翻訳しやすい日本語のペアを抽出し,このデータで BART をファインチューニングすることで,日本語文から翻訳しやすい日本語への変換器を構築する. 本手法の実現にあたり,取り組むべき課題は以下の 2 点である.

日本語文-翻訳しやすい日本語文のペアの取得

日本語文から翻訳しやすい日本語への日本語変換器を構築するために, 日本語文と翻訳しやすい日本語文のペアが必要である．日本語文は変換前の翻訳しにくい日本語でなければならない．翻訳しにくい日本語は機械翻訳によって日本人英語が生成される日本語文であるため, 日本語文を一度機械翻訳によって翻訳しなければならず, 大規模なデータを構築するには, 翻訳コストが非常に高い．そこで, 翻訳コストを抑えるために, できる限り翻訳しにくい日本語と推定されるものに原文段階で限定しなければならない.

日本語文から翻訳しやすい日本語への日本語変換器

日本語文と翻訳しやすい日本語文のペアを用いて日本語変換器を構築する必要がある．しかしながら, 変換モデルを学習するには, 日本語文と翻訳しやすい日本語文のペアが大規模に必要となるため, 事前学習済みのモデルを用いなければならない.

以下, 本論文では, 第 2 章で関連研究と機械翻訳の事前編集について述べる. 第 3 章で英語分類器と機械翻訳における並行経路について述べる．次に, 第 4 章で提案した日本語と翻訳しやすい日本語の抽出手法と使ったデータコーパスを紹介する．第 5 章で日本語を翻訳しやすい日本語に変換できる変換モデルの構築について紹介する．続いて, 第 6 章でそれぞれのコーパスで各バージョンの変換モデルの性能について述べる．最後に今後の発展と本研究の振り返りを述べる.

第2章 機械翻訳の事前編集

機械翻訳は現代社会において非常に広く応用されており,多くの利用者が存在する.現在,Google 翻訳や DeepL など,多くの高品質な機械翻訳ツールが日常生活や仕事の中で広く利用されている.これらのツールは多くの場合で満足のいく翻訳品質を提供するが,完全に正確かつ流暢な訳文を生成することは常に保証されているわけではない.翻訳効果を向上させるために,人々は通常,機械翻訳が生成した訳文を編集する.この編集過程は「事後編集」（Post-editing）と呼ばれ,機械翻訳の訳文を人工的に調整し,特定の応用場面により適したものにする.Štajner らはテキストを簡略化することで機械翻訳の流暢性などの指標を改善した[1].Killman らは英語-スペイン語の法律文書の翻訳における事後編集と機械翻訳の効果を研究し,事後編集が訳文の正確性と可読性を顕著に向上させることを発見した[2].De Almeida らは,機械翻訳の事後編集効果を詳しく分析し,異なるタイプの編集操作が訳文の品質にどのように影響するかを探討した[3].

事後編集に対するのが「事前編集」（Pre-editing）であり,これは翻訳前に原文を編集し,機械翻訳システムがより正確に翻訳しやすくすることを目的とする.事前編集の目的は,原文を簡略化および標準化することで,機械翻訳過程で発生する可能性のある誤りを減少させ,翻訳品質を向上させることである.Mercader-Alarcón らは,ルールを使用して機械翻訳を事前編集する方法を提案し,この方法によって機械翻訳の効果を顕著に向上させることができると示した[4].彼らの研究は,翻訳前に原文を適切に調整することで,訳文の正確性と流暢性が大幅に改善されることを示している.図 1 は,事前編集と事後編集が機械翻訳のどの段階で機能するかを示している.

機械翻訳の効果を判断する基準は多種多様であるが,本研究では英語の母語化程度を基準として使用する.英語文の母語化程度をどのように判断するかについては第三章で紹介する.本研究は,ニューラルネットワークを使用して日英翻訳タスクにおける事前編集を行うことである.私たちはデータを収集および生成し,BART モデルをファインチューニングして,日本語文原文を事前編集できるモデルを構築する.このモデルの目標は日本語文原文を機械翻訳がより正確に翻訳しやすくすることであり,最終的な訳文の品質を向上させることである.既存研究のレビューを通じて,事前編集が機械翻訳においてあまり応用されていないこ

とが分かった. 本研究はこの分野の空白を埋め, 日英翻訳タスクに新しいソリューションを提供する.

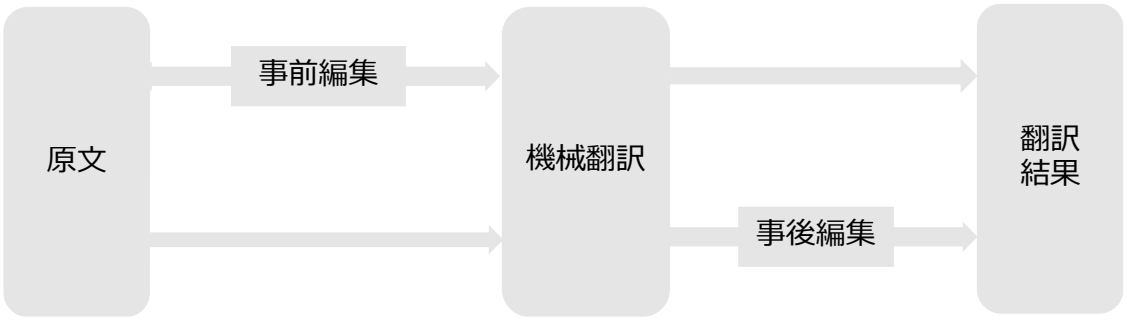

図 1 : 機械翻訳の事前編集と事後編集

第3章 機械翻訳における英語スタイル

英語のスタイルを同定するために,本章では英語のスタイル分類器の構築方法と分類器の訓練に必要となる訓練用のコーパスについて説明する.

3.1 英語スタイル分類器

英語のスタイル分類器はBERTを用いた分類器モデルである. BERTは,多くのNLPタスクにおいて革新的な結果を達成した Transformer[5]ベースの事前訓練済みモデルである[6]. その能力は深層双方向性とマルチヘッド注意力機構によるものである. BERT は文を理解するだけではなく,文脈も把握の能力もある. この能力を使って,英文スタイル分類器を構築する.

BERT事前訓練モデルは,大規模なデータコーパスから情報を抽出し,言語理解能力を備えた言語モデルである. BERTの訓練は,事前訓練とファインチューニングの2つのステップで行われる. 事前訓練では,モデルがMasked Language Modeling(MLM)とNext Sentence Prediction(NSP)という2つのタスクで訓練する. MLMは,BERTがマスク化された箇所の単語を予測する訓練タスクである. まずランダムに選択した文中の15%のトークンを[MASK]という特殊なトークンに置き換える. この新たに生成された文をBERTモデルに与え, [MASK]の位置に本来存在していたトークンを正確に予測するタスクを行う. このタスクは,[MASK]に置き換えられたトークンをそのトークンが存在するべき場所の「ラベル」または「正解」として扱うことで,モデルが単語の文脈に基づいた適切な関係を学習する. NSPは,BERTが2つの文の相互関連性を理解するための訓練タスクである. このタスクでは,BERTには必ず2つの文が一組として提供される. これらのペアのうち半分は,2つ目の文が1つ目の文の直後にくるもので,残りの半分は2つ目の文がランダムに選ばれている. そして,BERTはこれらの2つの文が連続したものかどうかを判断するタスクを用いて訓練を行う. 具体的には,特殊トークン[CLS]に対応するBERTの出力を分類器に入力し,その結果として2つの文が連続している（つまり,1つ目の文の直後に2つ目の文が来る）かどうかを判断する. このタスクを通じて,モデルは文間の関連性を理解する能力を習得できる.

BERT はトランスフォーマーをベースにしているが,トランスフォーマーのエンコーダ部分のみを使用し,トランスフォーマーの複数層のエンコーダを積み重ねて BERT を構築している.

図２は BERT の仕組みである．BERT の入力ベクトルは,3 種類のベクトル化方法の結果から組み合わされる．Token embedding は,単語をサブワードに分割し,次に各サブワードをベクトル化される,Token embedding は単語の意味ベクトル化のプロセスである．Position embedding は,各単語が文中のどの位置にあるかを表すベクトルを生成する．これにより,モデルは単語の意味だけでなく,その単語が文中のどの位置に存在するかという情報も学習することができる．Segment embedding は,各文またはセグメントに固有のベクトルを割り当てる,例えば,すべての「文１」のトークンには一つのベクトルが,すべての「文２」のトークンには別のベクトルが割り当てる．これにより,モデルは文ごとの区別が可能になり,文間の関係をより適切にモデル化することができる,NSP タスクで重要な役割を果たしている.

入力をベクトルに変換した後,N 層のエンコーダに入る,一般的に N は 12 である(N＝24 のモデルもある),エンコーダの各層は,各層に 12 個のアテンションがある,12 ヘッドアテンションのわけである．エンコーダは 1 層のマルチヘッドアテンションと 1 層のフィードフォワードから構成される．各アテンションの主な役割は,文中のすべての単語との相関によってターゲット単語を再コード化することである．　ターゲット単語は文中のすべての単語との関連性によって再コード化される．こうすることで,BERT は文の意味を理解するだけでなく,複数の意味を持つ文であっても理解することができ,RNN や LSTM と比べて,BERT モデルのロバスト性と汎用性がより高いである.

モデル内に残差連結という操作がある,残差連結は勾配消失または爆発といった問題を解決するためのものである,具体的には,ある層の入力が,その層の出力に直接加えられることで,勾配がネットワークを逆伝播するときに 0 に近づくまたは非常に大きくなる傾向があるという問題を解決する手段である．つまり,ネットワークは入力と出力の間の複雑な関数を直接学習する代わりに,その入力と出力の差を学習する．これにより,ネットワークの訓練が容易になり,また深いネットワークでもより良いパフォーマンスが得られることになる[7].

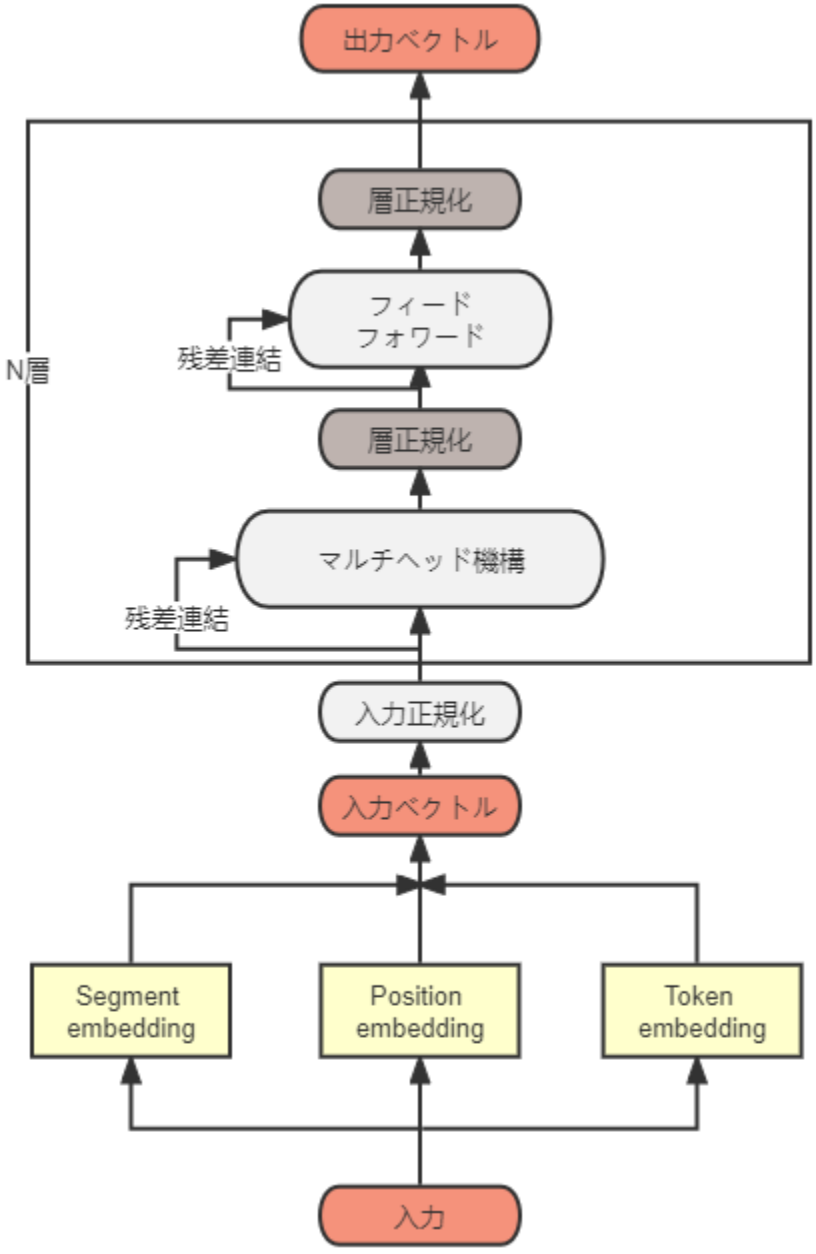

図２．BERT モデルの仕組み

3.1.1 英語スタイルの訓練データ

英語スタイル分類器を構築するために,学習データとして日本人書いた英語と英語母語話者書いた英語を収集し,ラベルを付けて,ラベルというのは,英語母語話者書いた英語にラベル０を付け,日本人書いた英語文にラベル１を付けてデータコーパスを構築する.

日本人書いた英語は「Wikipedia 日英京都関連文書対訳コーパス[1]」という人手翻訳コーパスを利用する,このコーパスの翻訳対象となった Wikipedia 記事は,京都に関する内容を中心に,日本の学校,鉄道,旧家,建造物,神道,人名,地名,伝統文化,道路,仏教,文学,役職と称号,歴史,神社仏閣,天皇という 15 分野をカバーしている. コーパスの翻訳は 3 ステップで行われました,一次翻訳は日本語を母語とする翻訳者が日本語原文を英訳する,二次翻訳は英語を母語とする翻訳者が一次翻訳文における情報の過不足および流暢さをチェックし,必要な場合は修正する. 三次翻訳は日本語を母語とするチェッカーが二次翻訳文における専門用語および言及する専門分野の知識のチェックを行い,必要な場合は修正する. 分類器の学習データは,日本人英語コーパスと母語話者英語コーパスの 2 つのデータコーパスから構成されている.

日本人英語コーパスは Wikipedia 日英京都関連文書対訳コーパスの一次翻訳の英語文を抽出してから,データを洗い出す. 具体的に,一次翻訳の文には括弧や非英語文字が多く含まれるため,モデルを学習する前にこれらの文字列をすべて削除する必要がある. これらの文字列を取り除かなければ,彼らはモデル分類の特徴となって,モデル学習の邪魔になりうる. そうなると,モデルは英語スタイルの分類器ではなく,特殊文字分類器になってしまう. そのうえに,データベースの品質を管理するため,長さ 3 以下の英文は削除する.

日本人英語コーパスの英文のトピックは日英京都関連文書対訳コーパスの 15 トピックなので,コーパス全体が偏らないようにするためには,母語話者英語コーパスの英文のトピックもその 15 トピックに含まれなければならない. そこで英語スタイル分類器の訓練データでは,日英京都関連文書対訳コーパスからすべてのウィキページのタイトルを抽出し,そのタイトルに従って対応する日本語ページを探し,そして,日本語のページから対応する英語のページにリダイレク

[1] https://alaginrc.nict.go.jp/WikiCorpus/

トし，見つかった英語のページはすべてクロールされ，母語話者英語コーパスを作成する．

3.1.2 英語スタイル分類器のファインチューニング

ファインチューニングはディープラーニングにおける一般的な手法で，特定のタスクに対して事前に訓練されたモデルをさらに調整することを指す．具体的には，大規模なデータセット上で訓練された深層学習モデルを，特定のタスクにより適したモデルに調整する．この調整は，通常，タスク固有の訓練データセット上でモデルの重みを微調整することによって行われる．ファインチューニングと事前訓練モデルの関係は，人間とツールの関係であり，モデルは，がどんな問題を解決したいなら，どんなツールを設計すればよい．

英語スタイル分類器の仕組みは図 3 のように示す，BERT の上に，線形層，活性化関数，線形層の順に 3 つの層が追加された．最後の線形層は，母語話者英語と日本人英語に分類される確率を表す 2 次元ベクトルを出力する．2 つの線形層と活性化関数を使うのは，1 つの線形層のみを使用するネットワークと比べて，モデルの複雑性，すなわち表現力が高い．データ中に存在する複雑で非線形なパターンをよりよく捉え，学習することができる．多くの場合，モデルがより良いパフォーマンスを提供するのに役立つ．活性化関数を導入することで，モデルは線形微分可能な境界だけでなく，より複雑な決定境界に適合することができ

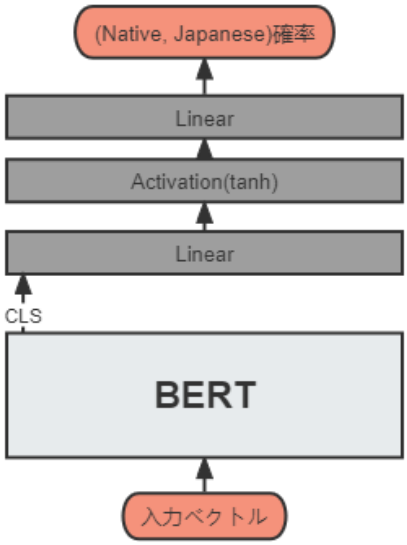

図 3:英語スタイル分類器の仕組み

る．実世界の多くのタスクでは非線形であるため，非線形性を導入することでモデルの汎用性がよくなれる．

3.1.3 分類モデルの信頼性

本研究では大量の部分に英語スタイル分類器を使用する必要があるため，英語スタイル分類器の信頼性を明確にすることが重要である．英語スタイル分類器の信頼性は，テストセットや ICNALE 学習者コーパスなどの異なるデータセットで検証されている．分類器は，訓練セットと同じ出所のテストセットにおける精度が高い．本研究で使用する英語スタイル分類器のテストセットでの正確率と F1 スコアは共に 90%以上に達している．そして，ICNALE 学習者コーパスに含まれる日本人が手書きした英語に対する判断の正確率も 90%以上である．これにより，本研究で使用する英語スタイル分類器は日本人英語と英語母語話者の英語を比較的正確に分類できることが示される．英語スタイル分類器の信頼性は保証される．分類器の F1 スコアと F2 スコアと検証結果はと図４と図５と表１になる．

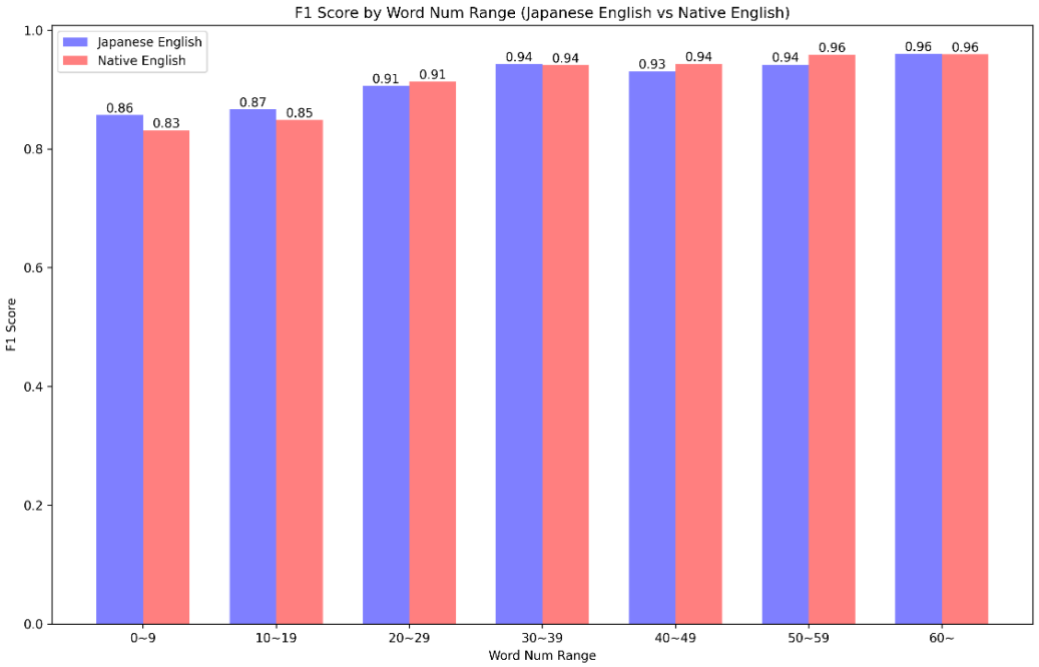

図４：分類器の F１スコア

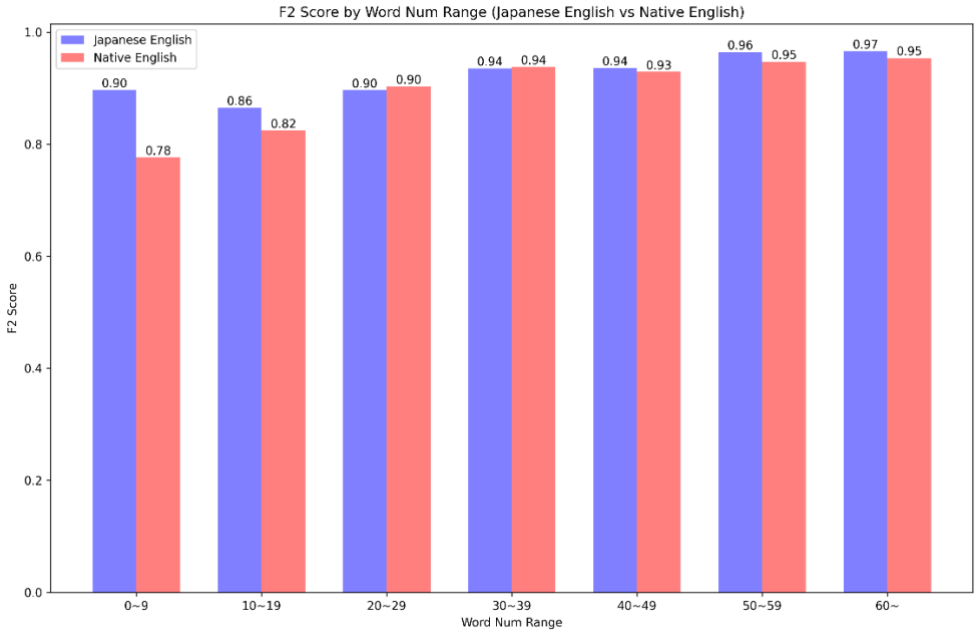

図 5：分類器の F2 スコア

表 1． ICJIA におけるエッセイの分類結果

	日本人英語判定数	割合
SAT エッセイ	73	15.3%
日本人学者エッセイ	11929	90.9%

3.2 機械翻訳における並行経路

機械翻訳が日英翻訳時にどのようなデータで高い母語化程度の英語を生成できるかを探究するため，事前実験を行った．具体的な実験手順は以下の通りである．まず，3.1 節の分類器のテスト集からランダムに選んだ日本人英語と母語話者英語それぞれ 3000 件を使用し，2 つのテストデータセットを構成した．元のデータが日本人英語のデータセットは「test-ja」と呼ばれ，元のデータが母語話者英語のデータセットは「test-en」と呼ばれる．

実験の第一歩として，これら 2 つのデータセットのテキストを全て日本語に翻訳した．次に，得られた 2 つの日本語翻訳集合を再度英語に翻訳し，2 つの新し

い機械翻訳後の英語データセットを生成した. こうして, 日本人英語から翻訳されたデータセットは以下に「test-ja-retrans」と呼ばれ, 母語話者英語から翻訳されたデータセットは「test-en-retrans」と呼ばれる.

これらの再翻訳されたデータセットの品質を評価するために, 3.1 節で言及した分類器を用いて母語化程度を判断した. 具体的には, データセットの母語化程度は各テキストが母語話者英語に属する確率と日本人英語に属する確率を計算し, それぞれを合計して平均値を求めることで決定した. この 2 つの平均値をデータセットの母語化程度の指標とした. 表 1 は, test-ja-retrans と test-en-retrans の 2 つのデータセットの母語化程度の結果を示している.

表 2： 3000 件の日本人英語と母語話者英語の逆翻訳の母語化程度

データセット	母語話者英語の確率	日本人英語の確率
test-ja-retrans	0.21626	0.78374
test-en-retrans	0.69046	0.30954

表 2 の結果から, 日本人英語は二度の翻訳後（英語から日本語, 再び英語へ翻訳）も依然として日本人英語の特徴を強く持っており, 母語話者英語の確率は 0.21626 であるのに対し, 日本人英語の確率は 0.78374 であることがわかる. これは, 二度の翻訳を経ても, 元の日本人英語が完全に母語話者英語に変換されていないことを示している. これに対し, 母語話者英語は同様に二度の翻訳後も依然として母語話者英語の特徴を強く保持しており, 母語話者英語の確率は 0.69046, 日本人英語の確率は 0.30954 である. この結果は, 母語話者英語が翻訳過程において高い一貫性を保っていることを示している.

この発見は重要な問題を表している. すなわち, 機械翻訳には並行経路が存在し, 全ての種類の日本語テキストを母語話者英語に翻訳することができないということである. 言い換えれば, 現在の機械翻訳システムは異なる種類の日本語テキストを処理する際に, 明らかな差異を示している. 特に, 元々日本人によって書かれた英語に関しては, 二度の翻訳を経ても, その翻訳結果は母語話者英語の水準に達していない. これは, 現行の機械翻訳システムが多様な日本語テキストを処理する際に, 明確な限界があることを意味している. 機械翻訳の並行経路の

イメージは図４になる. オレンジ色と緑色の矩形はそれぞれ日本人の経路と母語者の経路を表す.

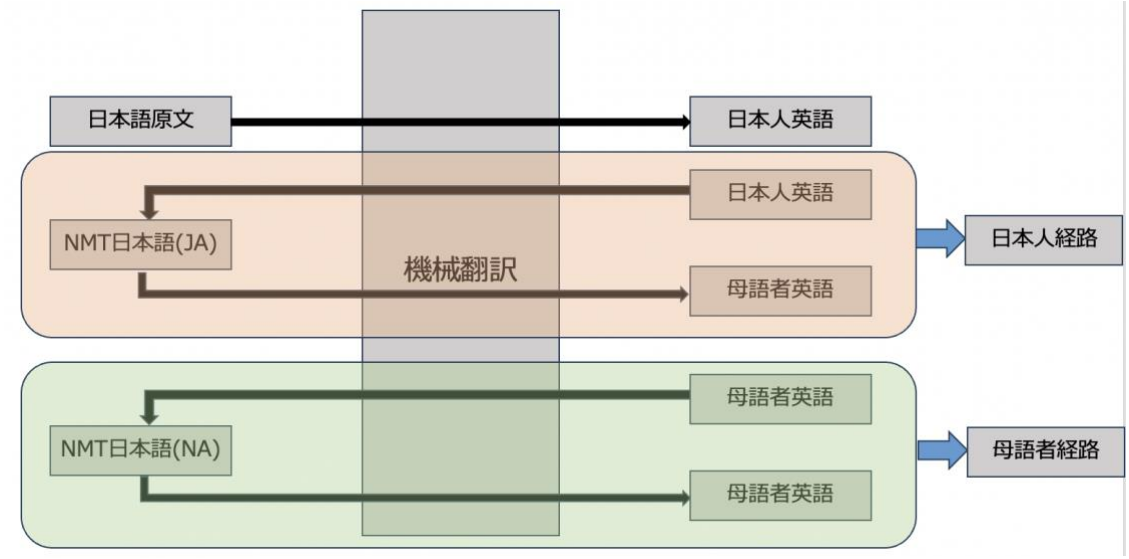

図 6:機械翻訳における並行経路

この問題を克服するために, 本研究では事前編集を提案する. それは, 事前編集を通じて, 母語話者英語に翻訳できない日本語テキストを, より翻訳しやすい形式に編集する方法である. 事前編集の核心は, ニューラルネットワークで原文を標準化することで, 機械翻訳過程で発生しうる誤りを減少させることである. この方法は, 翻訳の正確性と流暢性を向上させるだけでなく, 複雑なテキストを処理する際の機械翻訳システムの限界を効果的に解決することができる. 事前編集の流れは図７のようになる. 図７は, 事前編集が機械翻訳プロセスにおいてどの段階で作用するかを示している. 赤い矢印の部分は事前編集のプロセスを特に強調しており, 事前編集を通じて, 日本人経路上の日本語原文を英語母語者経路上の日本語に変換する. この変換により, 機械翻訳システムは日本語テキストをより効果的に母語話者英語に翻訳できるようになり, 翻訳の母語化程度と全体的な品質が向上する.

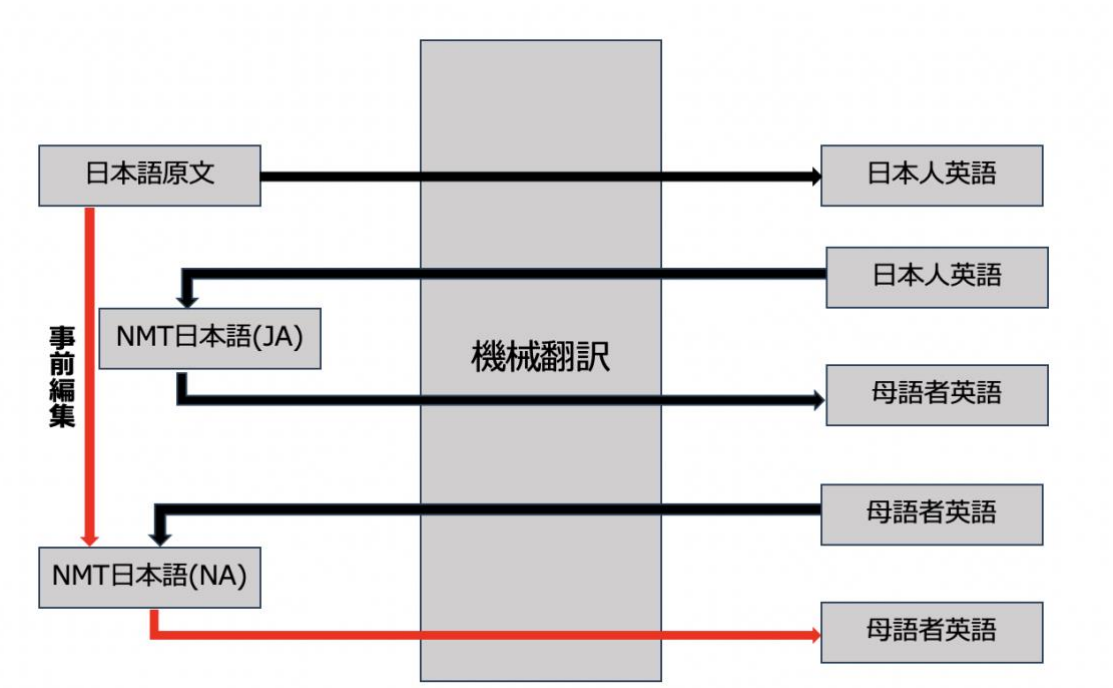

図 7:事前編集の流れ

本研究では「翻訳しやすい日本語」の定義が非常に重要である. 本研究において, 翻訳しやすい日本語とは, 機械翻訳を経て母語話者英語を生成できる日本語テキストを指す. 第 3 章第 2 節の並行経路仮説に基づけば, 母語者の英語が機械翻訳により生成された日本語テキストは, 再度機械翻訳を通じて母語話者英語を生成できることがわかる. したがって, 本研究では, 母語者の英語が機械翻訳によって生成された日本語を翻訳しやすい日本語とみなす.

第4章 対訳ペアの抽出

本章では,日本語と母語話者英語の対訳データが,機械翻訳に基づく並行経路理論において,日本語テキストと翻訳しやすい日本語の対の生成における重要性と抽出方法について詳しく探討する.まず,日本語テキストと翻訳しやすい日本語の対を収集する目的を紹介し,それが機械翻訳の品質向上において果たす重要な役割を説明する.次に,どのようにしてWikipediaの日英版から対訳データを抽出するかを紹介し,対訳データの選択とマッチングの方法と手順について述べる.

さらに,日本語テキストの翻訳難易度分類器を利用してテキストを分類し,翻訳しやすい日本語テキストをより正確に選別する方法についても探討する.次に,日本語翻訳難易度分類器を用いて,大規模な日英コーパスから対訳データを抽出する方法を詳述する.これには,データ構造,具体的な抽出方法,データのクレンジングおよび処理の過程が含まれる.

これらの一連の方法を通じて,高品質な日本語と母語話者英語の対訳データセットを構築することを目指す.このデータセットは,機械翻訳に基づく並行経路理論において,日本語テキストと翻訳しやすい日本語の対の生成に対する確固たるデータ基盤を提供する.このデータセットは,機械翻訳の正確性と流暢性の向上に寄与するだけでなく,事前編集方法の有効性をさらに検証するための基盤となる.最後に,これらのデータを使用してBARTモデルをファインチューニングし,事前翻訳モデルを構築することで,英語訳文の母語化程度を向上させる.

4.1 日本語文と翻訳しやすい日本語のペア

この節では,次章の日語変換モデルのためのデータ準備について詳述する.具体的には,日本語文と翻訳しやすい日本語のペアの構築方法である.これらのデータペアは,BARTモデルのファインチューニングに使用され,高品質な事前翻訳の実現を目指す.

高品質な日本語文と翻訳しやすい日本語のペアを得るために,まず日本語と母語話者英語の対訳文を収集する必要がある.これらの対訳文は,翻訳モデルの構築と事前翻訳研究の基礎となる.次のいくつかの小節では,これらの対訳文を異なるソースから取得する方法を詳述する.

具体的には,Wikipedia の日英版から対訳データペアを抽出する方法を探討する.Wikipedia は,オープンで内容が豊富なリソースであり,多言語対訳データを大量に提供しており,本研究に強力な基盤を提供する.

さらに,翻訳難易度分類器を用いて日本語テキストを分類し,翻訳に適した日本語テキストを選別する方法についても紹介する.この分類器は,大規模日英コーパスから適切な対訳文を選別するために使用される.

次に,大規模日英コーパスから対訳データペアを抽出する方法を紹介する.大規模日英コーパスは豊富な言語データを含んでおり,これらのデータの分析と処理を通じて,高品質な日英対訳文を取得することができる.これらの対訳文は,BART モデルのトレーニングに必要なデータサポートを提供する.

これらの方法を通じて,本研究は高品質な日本語文と翻訳しやすい日本語の対訳データセットを構築することを目指している.このデータセットは,本研究の核心目標である事前編集を通じて機械翻訳の品質を向上させるための堅実なデータ基盤を提供する.以下の各小節では,それぞれのデータソースと具体的な抽出方法について詳述する. 本研究の日本語原文と翻訳しやすい日本語のペアの作り方は図 8 のようになる.

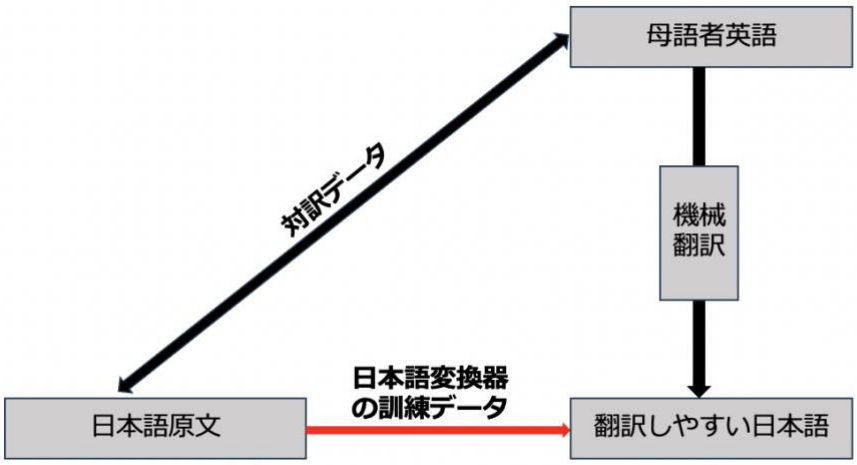

図 8：日本語原文と翻訳しやすい日本語の作り方

4.2 Wikipedia から日本語と母語話者英語の対訳ペアの抽出

Wikipedia（ウィキペディア）は,多言語の自由な内容を持つオンライン百科事典であり,ボランティアによって共同で執筆および維持されている.2001 年に設立されて以来,ウィキペディアは世界最大かつ最も包括的な参考資料の一つとなっている.ウィキペディアの項目は,科学,歴史,文化,技術など,多岐にわたるテーマを網羅している.ウィキペディアのオープン性とグローバルな協力の特徴から,その内容は非常に豊富で多様であり,常に更新されている.

ウィキペディアのコーパスは非常に豊富であり,特に多言語の対訳において重要な資源となっている.各項目には通常,複数の言語バージョンがあり,これにより日英の対訳文を抽出するための貴重な資源が提供されている.ウィキペディアの日語版と英語版からは,大量の対訳文を見つけることができ,これらの対訳文の品質と一貫性は,本研究に非常に有益である.

ウィキペディアから日語と英語の対訳文を抽出することで,高品質な日英対訳データセットを構築することができる.このデータセットは,本研究に堅固な基盤を提供するだけでなく,機械翻訳モデルの訓練および検証にも使用することができる.具体的な抽出方法およびその原理については,本節の後続部分で詳しく説明する.

4.2.1 Wikipedia の日本語版と英語版

ウィキペディアの大部分の項目には複数の言語バージョンが存在する.これらの言語バージョンは内容が似ているものの,通常は独立して執筆されており,直接的な関連性は少ない.しかし,日本文化に特有の概念や出来事については,項目を執筆する際に母語者が日本語版の内容を参考にして英語版を作成することが多い.

このような場合,内容が高度に一致するページを見つけ出し,そこから日本語と母語話者英語の間の対訳文を抽出することができる.例えば,日語版ウィキペディアの「神饌」の項目の「形式」部分と,英語版ウィキペディアの「Shinsen」の項目の「Process」部分は内容が高度に一致している.こうしたページを利用することで,高品質な対訳文を得ることができ,これらの対訳文は本研究およびモデル訓練に非常に貴重である.

本研究では,ウィキペディアからこのようなページをさらに多く見つけ出し,これらのページから高度に一致する日本語と英語の内容を抽出することで,高品質な日英対訳データセットを構築することを目指している.内容が高度に一致する日本語版と英語版のウィキペディアページの例を図 9 に示す.具体的な抽出方法およびその原理については,本節の後続部分で詳しく説明する.

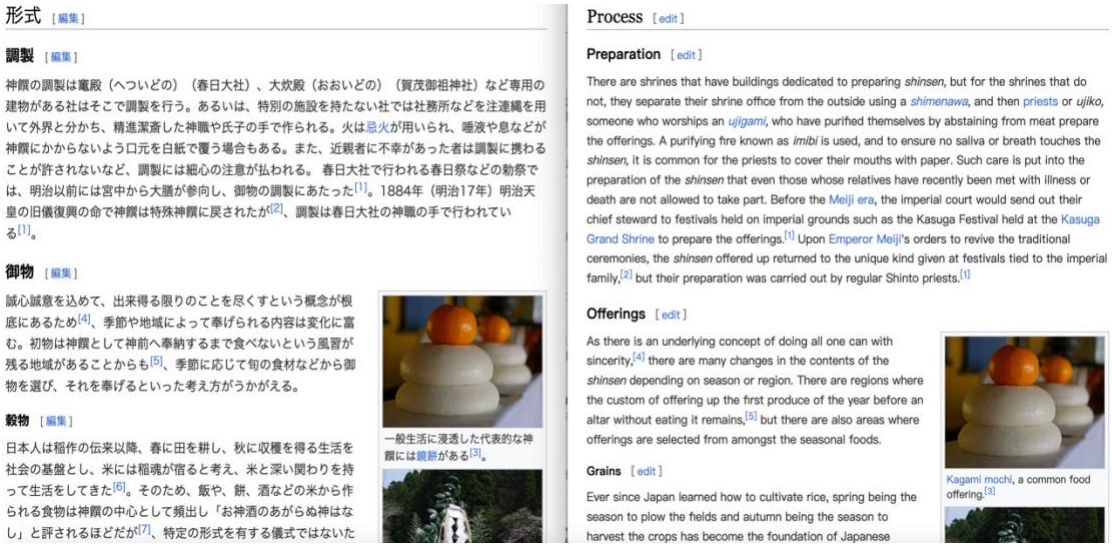

図 9:高度に一致する日本語と英語の内容

4.2.2 テキスト埋め込みモデル

この小節では,Sentence-BERT（SBERT）[8]について詳述する.SBERT は,Siamese および三重項ネットワーク構造を用いて,事前学習された BERT ネットワークを修正し,意味的に有意義な文の埋め込みを生成する方法である.

BERT（Devlin et al., 2018）および RoBERTa（Liu et al., 2019）は,文ペア回帰タスク（例えば,意味的テキスト類似度タスク）において新しい性能基準を設定した.しかし,これらのモデルは二つの文を同時にネットワークに入力する必要があり,これが大きな計算オーバーヘッドを引き起こす.例えば,10,000 文を含む集合で最も類似したペアを見つけるには約 5000 万回の推論計算（約 65 時間）が必要となる.この構造により,BERT は意味的類似性検索や無監督タスク（例えば,クラスタリング）には不向きである.これらの問題を解決するために,Sentence-BERT（SBERT）が提案された.SBERT は,Siamese および三重項ネットワーク構造を用いることで,意味的に有意義な文の埋め込みを生成し,これらの埋め込みを余弦類似度を用いて比較することができる.この方法により,最も

類似した文を見つける時間を 65 時間から約 5 秒に短縮し,BERT の精度を維持することができる.

SBERT は,BERT/RoBERTa の出力にプーリング操作を追加し,固定サイズの文の埋め込みを得る.著者は三つのプーリング戦略を試みた：CLS トークンの出力を使用する,全ての出力ベクトルの平均を計算する（MEAN 戦略）,出力ベクトルの最大値を計算する（MAX 戦略）.デフォルトの設定は MEAN である.

BERT/RoBERTa を微調整するために,著者は Siamese および三重項ネットワークを作成し,重みを更新して,生成される文の埋め込みが意味的に有意義であり,余弦類似度を用いて比較できるようにした.図 10 は SBERT の分類目的関数の仕組みを示し,図 11 は推論時の仕組みを示している.

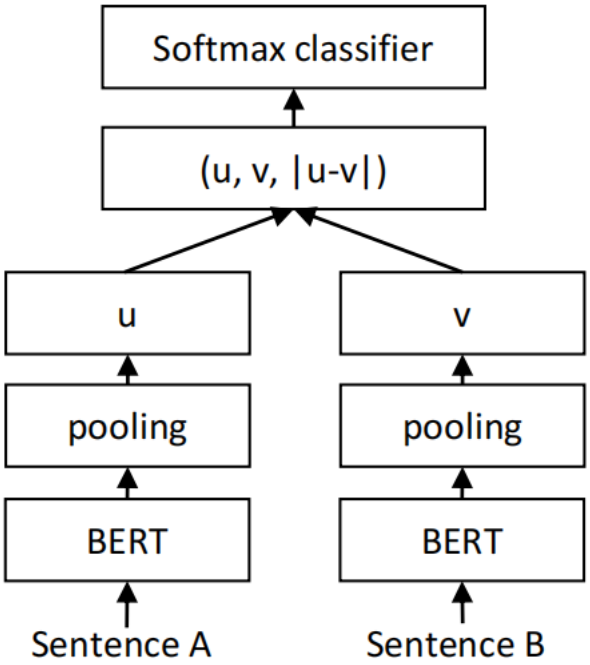

図 10: SBERT の分類目的関数の仕組み

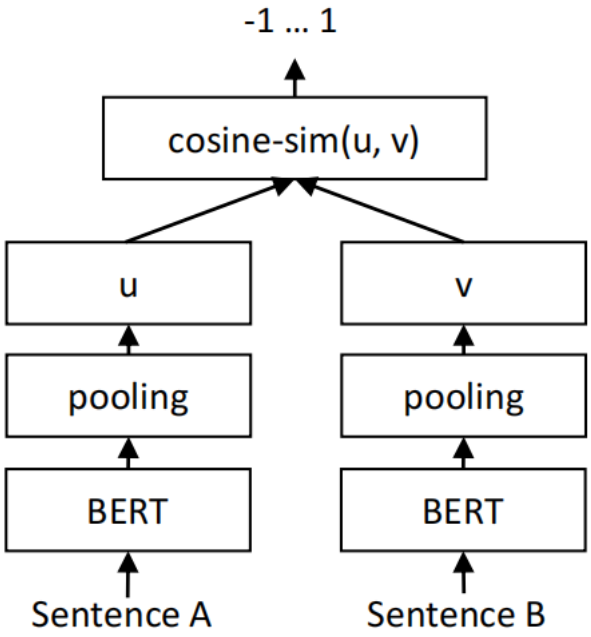

図 11: SBERT の推理時の仕組み

SBERT の応用と利点は以下の通りである.SBERT は,一般的な意味テキスト類似度（STS）タスクおよび転移学習タスクにおいて優れた性能を発揮し,他の最新の文埋め込み方法を上回っている.SBERT は,InferSent および Universal Sentence Encoder と比較して,七つの STS タスクで平均 11.7%および 5.5%の改善を達成した.

SBERT の効率性は,計算の複雑さの大幅な削減にある.SBERT を使用すると,10,000 文の中で最も類似したペアを見つけるために,10,000 文の埋め込みを計算するだけで済み（約 5 秒）,余弦類似度の計算も約 0.01 秒で済む.最適化されたインデックス構造を使用することで,最も類似した Quora の質問を見つける時間が 50 時間から数ミリ秒に減少した.さらに,SBERT は特定のタスクに適応させることができ,挑戦的なデータセットにおいて新しい性能基準を設定している.例えば,論点類似性データセットや Wikipedia の章区別データセットなどが挙げられる.

SBERT を導入することで,Wikipedia から日本語と母語話者英語の対訳文を効果的に抽出できるようになり,本研究に確固たるデータ基盤を提供する.この具体的な抽出方法と原理については,本節の後続部分で詳しく説明する.

4.2.3 Wikipedia から日本語と母語話者英語の対訳ペア抽出手法

この節では,Wikipedia から日本語と母語話者英語の文対を抽出する具体的な手順について詳しく説明する.

ステップ 1：クロールで Wiki ページから文章を取得する.具体的に,ウェブクローリング技術を使用して Wikipedia ページから文章内容を取得する.このステップの目的は,後続の処理に必要な関連テキストをできるだけ多く収集することである.クローラーを使用することで,Wikipedia の日語ページと英語ページを体系的に巡回し,完全な段落と文を抽出する.

ステップ 2:ステップ 1 で取得した段落文章を単文にする.具体的に,取得した Wikipedia ページの段落テキストを単文に分解する.この方法により,日本語と英語の文対をより正確に処理および比較することが可能になる.このステップの目的は,後続のステップで文単位のマッチングと比較をより良く行うためである.

ステップ 3：ステップ 2 で取得した日本語単文を Google 翻訳で翻訳して「Sentences-BERT」で文の埋め込みを取得する.具体的に,ステップ 2 で取得した日本語単文を Google 翻訳で英語に翻訳する.その後,Sentence-BERT (SBERT) を使用してこれらの翻訳文を埋め込み処理する.SBERT は,意味的に有意義な文の埋め込みを生成し,後続の類似度計算を容易にする.

ステップ 4:日本語文の一文を複数の英語文に対応できるようにするために window_size(1,2,3,4,5) で英語文を段落にした上記の新しい英語文を「Sentences-BERT」で文の埋め込みを取得する.具体的に, window_size (1,2,3,4,5) を用いてステップ２で取得した英語文を段落にする.その後,新しい英語段落を SBERT で各段落の埋め込みを取得する.この方法により,一つの日本語文にとって複数の英語文との類似度を比較でき,複数の英語文に対応できるようになった.

ステップ 5:各日本語文に対する cos 類似度が一番高いかつ cos 類似度は 0.8 以上の英語文を見つける.具体的に,すべての文の埋め込みを取得した後,各日本語文とすべての英語文の余弦類似度を計算する.次に,各日本語文に対して最も高い類似度を持ち,かつ類似度が 0.8 を超える英語文を見つける.このようにして,選択された文対が意味的に高い一致性を持つことを確保する.

ステップ 6：ステップ 5 の結果から日語と対応する英語の文のペアを生成する.具体的に,ステップ 5 の結果に基づき,最も高い類似度を持ち条件を満たす日本語文と英語文を文対として組み合わせる.これらの文対は,日本語と母語話者

英語の対訳データセットの重要な部分となり,後続の研究およびモデル訓練の基礎を提供する.

以上の手順を通じて,Wikipedia から高品質な日本語と母語話者英語の文対を体系的に抽出することができる.抽出の手順は図 12 にようになる.

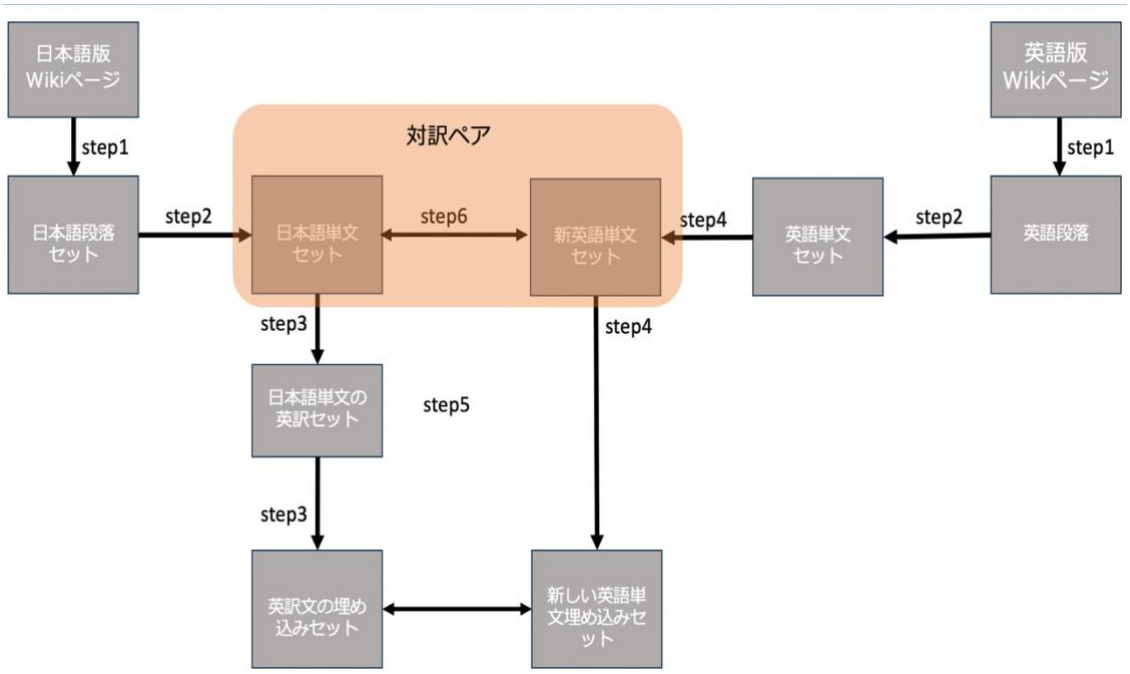

図 12: Wikipedia から対訳ペアを抽出する手順

手作業でウィキペディアのページをチェックすることにより,本研究では表 3 に示すような日本語ページと対応する英語ページを見つけた.上記の抽出方法を通じて,これらのページから 700 件の高品質な対訳ペアを抽出した.これらの翻訳ペアは意味的に高度に一致しており,次の研究ステップに向けた堅実なデータ基盤を提供した.

表３：対訳ペアを抽出できる Wiki ページ

日本語版のページの名前	英語版のページの名前
陰陽師	Onmyōji
果心居士	Koji Kashin
神饌	Shinsen
山岳信仰	Mountain worship
産土神	Ubusunagami
月読神社 (京都市)	Tsukiyomi Shrine (Kyoto)
天津神・国津神	Amatsukami and Kunitsukami
神祇院	Institute of Divinities
神社合祀	Shrine Consolidation Policy
氏神	Ujigami
講	Kō
鎮守神	Chinjugami
生祀	Worship of the living
大祓詞	Oharae no Kotoba
鯨塚	Whale mounds
鎮守の森	Chinju no Mori
三方 (神道)	Sanbo
神木	Shinboku
皇典講究所	Office of Japanese Classics Research
忌部氏	Inbe clan
大教宣布	Taikyo Proclamation

4.3 日本語文の翻訳難易度分類器

本節では日本語翻訳難易度分類器を紹介し, 翻訳しやすい日本語と翻訳し難い日本語を定義する. 本研究の日本語翻訳難易度分類器の目的は, 日本語文が機械翻訳を経て, 母語話者英語になるか日本人英語になるかを判断することである. 次の節「大規模な日英コーパスから対訳ペアの抽出」で使用される. 本研究では, 機械翻訳を経て母語話者英語を生成する日本語文を翻訳しやすい日本語と呼び, 機械翻訳を経て日本人英語を生成する日本語文を翻訳し難い日本語と呼ぶ. この分類器の訓練データ, モデルの選択およびモデル性能については本節で詳しく説明する.

4.3.1 訓練データ

日本語翻訳難易度分類器を訓練するために,まず大量の日本語文を収集する必要がある.これには,ウィキペディアやインターネットから収集した日本語文,合計 10000 文を使用した.訓練データの多様性と広範性を確保するために,歴史,文化,科学,技術など,さまざまな分野とテーマの日本語文を選んだ.

これらの日本語文を収集した後,DeepL 翻訳サービスを使用してこれらの日本語文を英語に翻訳した.DeepL は現在最も先進的な機械翻訳サービスの一つである.DeepL 翻訳を通し,高品質の英語訳文を得た.次に,これらの英語訳文を英語スタイル分類器で分類した.この分類器は,英語文が母語話者英語か日本人英語かを判断することができる.英訳が母語話者英語と分類された日本語文にはラベル 0 を,日本人英語と分類された日本語文にはラベル 1 を付けた.この方法により,最終的に 2500 文の翻訳しやすい日本語と 7500 文の翻訳し難い日本語が得られた.これらのラベル付きデータは,日本語翻訳難易度分類器を訓練するための堅実な基盤を提供する.これらのデータを使用して,機械翻訳を経た英語文のタイプを予測し,その日本語文の翻訳難易度を判断するモデルを訓練することができる.日本語翻訳難易度分類器の訓練データを作る流れは図 13 の通りである.

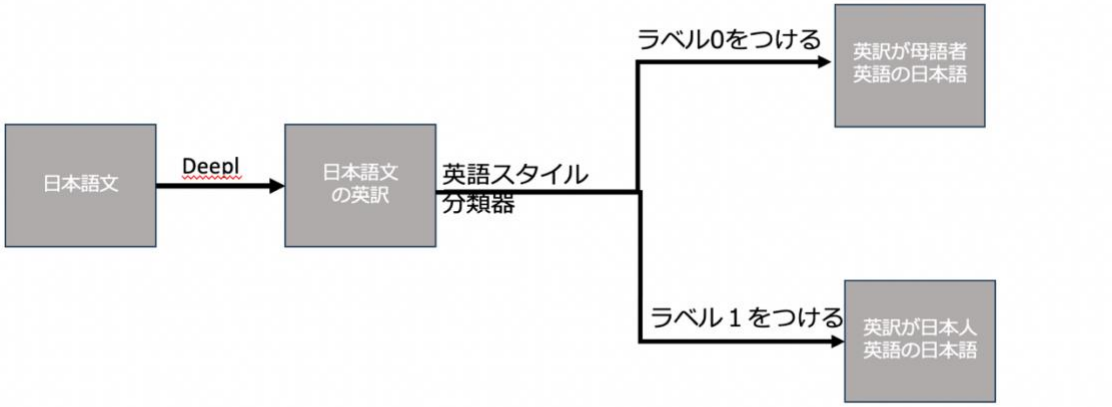

図 13: 日本語翻訳難易度分類器の訓練データを作る流れ

4.3.2 BERT モデルを用いた日本語文の翻訳難易度分類器と性能分析

BERT (Bidirectional Encoder Representations from Transformers) モデルの自然言語処理分野での成功に伴い,多くの言語に対して独自の事前訓練モデルが登場している.日本語も例外ではない.日本語の BERT 事前訓練モデルは日本東北大学 (Tohoku University) によって開発され,Hugging Face プラットフォー

ムで公開されている.このモデルは日本版 Wikipedia と CC-100 データセットを含む複数の日本語データセットで訓練されている.具体的には,Wikipedia Cirrussearch のテキストデータと CC-100 データセットの日本語部分を使用しており,約 426M 文を含んでいる.

日本版 BERT モデル（tohoku-nlp/bert-base-japanese）は,元の BERT モデルのアーキテクチャに基づいており,12 層,768 次元の隠れ状態,および 12 のアテンションヘッドを持つ.このモデルの前処理には MeCab 形態素解析器と Unidic 2.1.2 辞書を用いて分かち書きを行い,WordPiece アルゴリズムでサブワード分割を行い,語彙表のサイズは 32768 である.

訓練データには日本版 Wikipedia と CC-100 データセットが含まれており,前者のテキストコーパスのサイズは 4.9GB,約 34M 文を含み,後者のコーパスサイズは 74.3GB,約 392M 文を含む.訓練中には Google 提供の Cloud TPU（v3-8 インスタンス）を使用し,マスク言語モデル（MLM）目標を採用し,全語マスクを有効にした.

本研究では,上記の日語 BERT 事前訓練モデルと,3.1 節で得られた 10000 件のデータを用いてファインチューニングを行った.ファインチューニングの過程では,これらのデータを事前訓練モデルに入力し,モデルのパラメータをさらに調整して,特定のタスク,すなわち日本語文の翻訳難易度を判断することにより適応させる.

具体的には,訓練データセットには 2500 件の翻訳しやすい日本語文と 7500 件の翻訳し難い日本語文が含まれている.これらのデータを日本語 BERT モデルに入力し,複数回の反復訓練を行った.各反復において,モデルは入力データに基づいて内部パラメータを調整し,日本語文の翻訳難易度をより正確に分類できるようにする.

ファインチューニング後,以下の性能指標が得られた：

精度（Accuracy）：0.59

日本人英語文の再現率（Recall for Japanese English sentences）：0.3

日本人英語文の適合率（Precision for Japanese English sentences）：0.7

このモデルの精度は理想的なレベルには達していないものの,大規模データセットにおけるスクリーニング機能は依然として非常に重要である.この分類器を通じて,翻訳しやすい日本語文と翻訳し難い日本語文を効果的に区別することができ,後続の研究に強力なサポートを提供する.

総じて, 日本語 BERT 事前訓練モデルの本研究への応用は, 日本語テキスト処理の強力な能力を示している. モデルの精度には改善の余地があるが, その分類タスクにおける性能は本研究ニーズを十分に満たしており, 次のステップへの堅実な基盤を築いている. 日本語文の翻訳難易度分類器のイメージは図 14 のようになる.

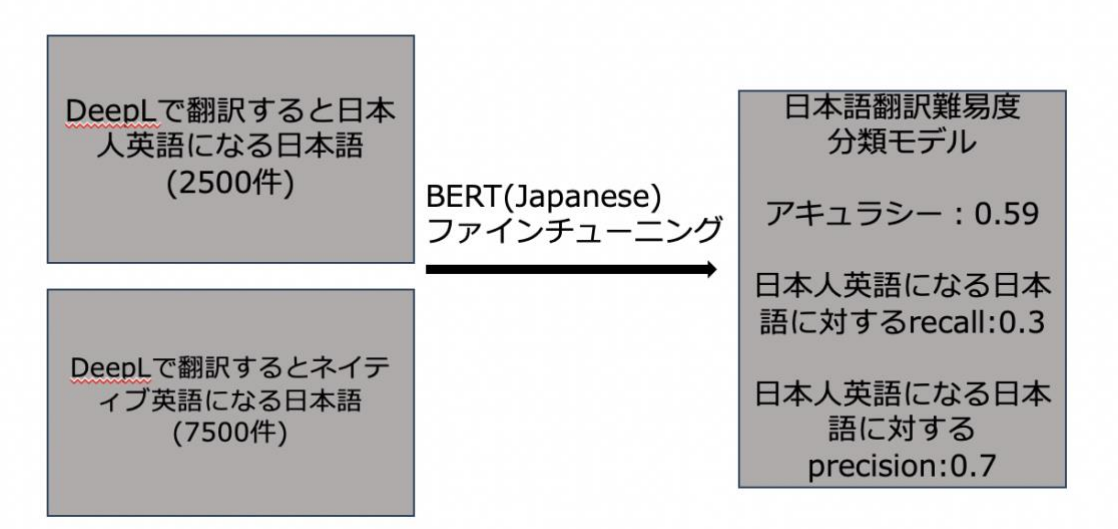

図 14: 日本語翻訳難易度分類器の訓練データを作る流れ

4.4 大規模日英コーパスから日本語と母語話者英語の対訳ペア抽出

大規模な日英対訳コーパスは, 多くの日語文と英語文からなるデータベースであり, これらの対訳コーパスは通常, 文学作品, ニュース記事, 技術文書, ウェブスクレイピングされた内容など, さまざまなテキストリソースから得られる. これらの対訳コーパスの目的は, 機械翻訳, 自然言語処理, 言語学研究に豊富なデータリソースを提供することである. 本研究では, JParaCrawl というインターネットからスクレイピングされたテキストを組み合わせた日英対訳コーパスを使用する. JParaCrawl には, 高品質の日英対訳文が多数含まれており, 機械翻訳と対訳分析の研究にとって貴重なリソースである. JParaCrawl から日本語と母語話者英語の文ペアを抽出することで, 本研究に堅実なデータ基盤を提供する. ウィキペディアから文を抽出する場合とは異なり, JParaCrawl コーパスには豊富な日英対訳リソースが含まれているため, 今回の抽出では英語部分の高度な英語母

語化に加えて,日本語部分に基づいてさらにフィルタリングすることを目指している.具体的には,日本語-母語話者英語の文ペアに対して,翻訳し難い日本語と母語話者英語の文ペアを得るためにさらにフィルタリングを行い,後続のモデル訓練をより適切に指導する.この方法により,選択された文ペアが意味と文法の両方において高度な一貫性を持ち,モデルが翻訳し難い日本語と翻訳しやすい日本語の特徴をより正確に学習できるようにし,日本語テキストの事前編集をより良く行うことができる.この具体的な抽出方法および原理については,本節の後続で詳細に説明する.

4.4.1 大規模日英コーパスのデータ構造

本節では,本研究で使用した大規模日英対訳コーパス「JParaCrawl[9]」について詳しく説明する.JParaCrawl はウェブクローラを使用してインターネットから取得したテキストを組み合わせた日英対訳コーパスであり,機械翻訳,自然言語処理,言語学研究に豊富なデータリソースを提供することを目的としている.以下に JParaCrawl の詳細を紹介する.

JParaCrawl v3.0 はこのコーパスの最新バージョンであり,2100 万以上のユニークな文ペアを含む.前のバージョン（v1.0 および v2.0）と比較して,JParaCrawl v3.0 の文ペア数は 2 倍以上に増加した.JParaCrawl の構築プロセスには,以下の主要なステップが含まれる：

1. 並行テキストを含むサイトの特定：Common Crawl テキストアーカイブデータを分析し,同じ英語と日本語の文を含むサイトを特定し,約 10 万の英語と日本語のテキストを含む大規模なサイトをリストアップする.
2. これらのサイトをクロールする:Heritrix クロールツールを使用してこれらのサイトをクロールし,プレーンテキスト,PDF,および Microsoft Word 文書を含む多くの並行文を抽出する.
3. 並行文の抽出:Bitextor ツールを使用してクロールしたアーカイブから並行文を抽出し,bleualign ツールを使用して BLEU スコアを最大化し,最適な英語-日本語文ペアを見つけて整列させる.
4. ノイズ文のフィルタリング：Bicleaner[10]ツールを使用して,整列エラーや翻訳不良のノイズ文ペアをフィルタリングし,最終的に 2100 万以上の文ペアを持つ JParaCrawl v3.0 コーパスを構築する.

JParaCrawl のテキストファイルには,異なる情報を表す複数の列が含まれる.これらの列には以下が含まれる：

- テキストの出典ウェブサイト（Source Website）：日本語の原文を含む.
- 日本語原文（Source sentence）：日本語の原文を含む.
- 英語対訳文（Target sentence）：対応する英語文を含む.
- Bicleaner スコア（Bicleaner Score）：二つの文の意味的な類似度を示すスコア.Bicleaner スコアが高いほど,二つの文が意味的に近いことを示す.

これらの情報は文ペアの唯一性と整列の正確性を確保し,研究や応用に豊富なメタデータを提供する.JParaCrawl を通じて,大規模なデータセットから高品質な日英文ペアを抽出することができる.具体的には,日本語と母語話者英語の文ペアに対してさらにフィルタリングを行い,翻訳し難い日本語と母語話者英語の文ペアを得ることを目指している.これにより,後続のモデル訓練をより適切に指導することができる.JParaCrawl の豊富な言語資源は,機械翻訳および自然言語処理分野において広範な応用可能性を持ち,モデルの訓練効果と翻訳精度を効果的に向上させることができる.

4.4.2 大規模日英コーパスから日本語と母語話者英語のペア抽出手法

本節では,大規模なコーパスから必要な文ペアをどのように抽出するかを詳述する.ウィキペディアから文ペアを抽出する方法とは異なり,今回は既に整列された日本語と英語の文ペアをフィルタリングする必要がある.具体的なステップは以下の通りである.

4.4.2.1 ステップ 1：Bicleaner 値が 0.75 以上の文ペアを抽出する

Bicleanerは,並行文ペアの品質を評価するためのツールであり,Bicleaner値が高いほど,2つの文が意味的に近いことを示す.このステップでは,まずBicleaner値が0.75以上の文ペアを抽出する.観察によれば,Bicleaner値が0.75以上の日本語と英語の文は意味的に高度な一致を示すため,0.75をフィルタリング基準として選択した.このステップの目的は,意味的に高度な一致を持つ文ペアを確保し,後続のフィルタリングに高品質の基礎データを提供することである.本研究では,このステップで約778万文ペアを抽出した.Bicleaner値が0.75以上の文ペアを抽出することで,意味の一致を保証し,後続の処理の作業量を大幅に減らすことができる.

4.4.2.2 ステップ 2：日本語ー母語話者英語の対訳抽出

高品質の文ペアを取得した後,英語部分を英語スタイル分類器で分類する.英語スタイル分類器は,英語文が母語話者英語か非母語話者英語かを判断できる.こ

のステップの目的は,選択した英語文が母語話者の言語特徴を持つことを保証し,翻訳品質の評価基準を高めることである.このステップでは,母語話者英語の文ペアを抽出した.本研究では,118万件の母語話者英語文ペアを抽出した.この分類により,英語母語話者を含む日英ペアを選出できる.

4.4.2.3 ステップ 3：分類モデルを用いて日本語から翻訳しにくい日本語を抽出

翻訳し難い日本語を含む文ペアを抽出するためには,理論上118万文の日本語を翻訳する必要があるが,これは実際にはコストが高すぎる.そのため,前節で紹介した日本語翻訳難易度分類器を使用し,翻訳し難い日本語を含む可能性のある文ペアをフィルタリングする.このステップでは,これらの文ペアを「翻訳し難い日本語（推定）-母語話者英語文ペア」と呼ぶ.本研究では,このステップで63000件の「翻訳し難い日本語（推定）-母語話者英語文ペア」を抽出した.このステップの目的は,実際に翻訳する必要のある文の数を減らし,コストを削減し効率を向上させることである.日本語翻訳難易度分類器の適用により,機械翻訳過程で処理が難しい可能性のある日本語文をより正確に見つけることができる.

4.4.2.4 ステップ 4：不完全な日本語文を含む文ペアを削除する

ステップ３で抽出された文ペアの中には,不完全なテキストを含む日本語文がある可能性がある.データの品質を確保するために,不完全な日本語文を含む文ペアを削除した.不完全な日本語文の定義は,以下の記号を含む日本語文である：:;()!? ： ； （）！？「」一-€$ ＄ •[] ［］【】など.これらの記号を含む文は一般的ではないため事前編集のモデルに影響を与える可能性があるので削除するべきである.本研究では,このステップで13000件の文ペアを抽出し,このデータセットを「Jpara-big」と呼ぶ.不完全な日本語文を削除することで,文の意味の一貫性を保証し,データのノイズを削除できる.

4.4.2.5 ステップ 5：NMT を用いて，翻訳しにくい日本語―翻訳しやすい日本語の対訳ペアを抽出

ステップ４で得られた文ペアの中から,日本語部分をDeepLで翻訳し,次に英語スタイル分類器で訳文を分類する.訳文が日本人英語と分類された日本語原文の文ペアを抽出し,最終的な翻訳し難い日本語-母語話者英語文ペアを得る.本研究では,このステップで2386件の文ペアを抽出し,最終的なデータセットを「Jpara-small」と呼ぶ.

上記のステップを通じて,大規模なコーパスから高品質の日英文ペアを効果的に抽出することができる.これらの文ペアは意味的に高度に一致しているだけでなく,日本語変換モデルに合わせる翻訳し難い日本語（推定）-母語話者英語文ペアである.これらのデータは後のモデル訓練に堅実なデータ基盤を提供した.Jparaからデータを抽出する流れは図15の示すようである.

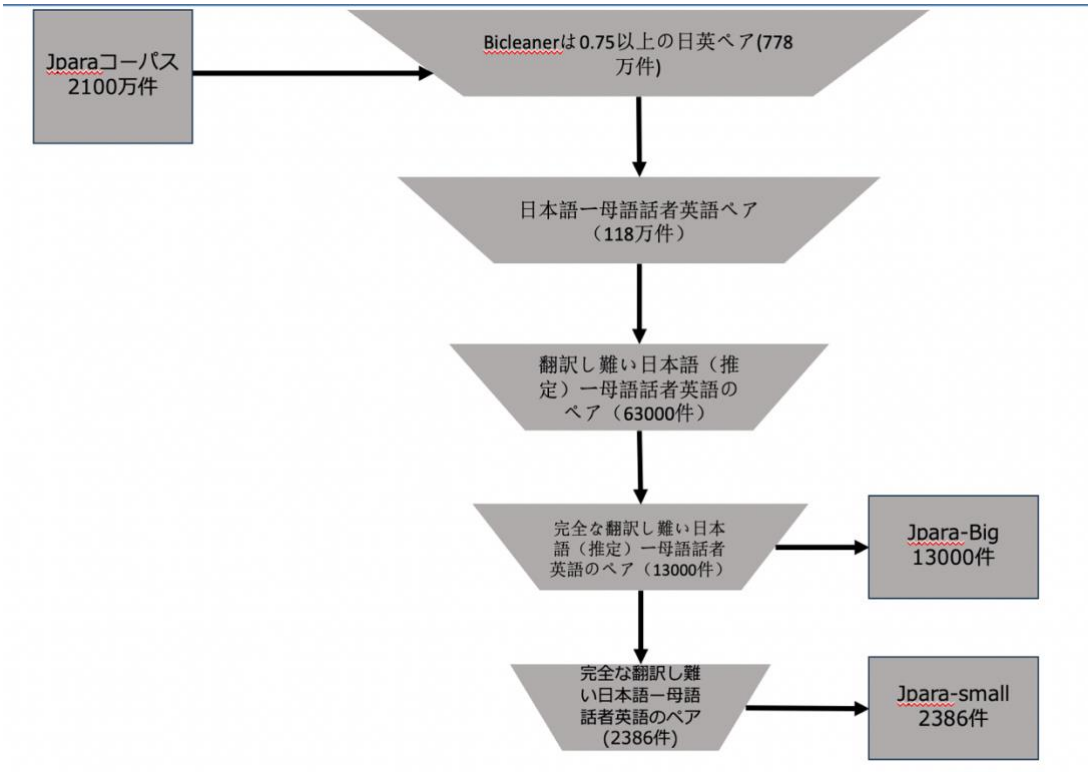

図 15: Jpara からデータを抽出する流れ

4.5 日本語と母語話者英語の対訳ペアの抽出結果

この節では, 本章の成果をまとめる. 詳細なステップとフィルタリングルールを通じて, 本研究では 3 つの高品質な日本語と母語話者英語の対訳データセットを構築し, 後続の日本語変換モデルのファインチューニングに堅実なデータ基盤を提供した.

まず, ウィキペディアのページを手動でチェックし, SBERT（Sentence-BERT）のルールを用いて, 多数のウィキペディアページから 700 の高品質な日本語と母語話者英語の文ペアを抽出した. そして, 小説という日英コーパスから日

本人著者と英語母語話者訳者の日英ペア 900 件を抽出した. それらを合併することで日本語と母語話者英語のデータセットを作った. これらの文ペアは意味的に高度な一致を持ち, 厳密なフィルタリングと検証を経ている. 本研究では, このデータセットを pairset-Wiki と呼ぶ.

次に, 大規模な日英翻訳コーパス（JParaCrawl）からのフィルタリングにより, 2386 の翻訳し難い日本語と母語話者英語の文ペアを成功に抽出した. これらの文ペアの品質を確保するために, Bicleaner 値を用いた初期フィルタリング, 英語スタイル分類器を用いた母語話者英語の文の選別, 日本語翻訳難易度分類器を用いた翻訳し難い日本語のフィルタリング, 不完全な日本語を含む文ペアの削除, そして最終的な深度フィルタリングの一連のルールを適用した. 本研究では, このデータセットを Jpara-small と呼ぶ.

さらに, 大規模な日英翻訳コーパスから, より多くのデータを含む日本語と母語話者英語の文ペアを 13000 抽出した. これらの文ペアは, 意味と文法の両方で高度な一致を持ち, 研究により豊富なデータ基盤を提供する. 本研究では, このデータセットを Jpara-big と呼ぶ.

pairset-Wiki, Jpara-small, および Jpara-big のデータセットを構築することにより, 後続の日本語変換モデルのファインチューニングに重要なリソースを提供した. これらの高品質なデータセットは, 機械翻訳の母語化程度を向上させるだけでなく, 並行経路理論をさらに検証し, 自然言語処理分野の研究に新たな見解と方法を提供する.

第5章 日本語変換器

この章では,前章で取得したデータを用いて,BART モデルのファインチューニングを行い,日本語から日本語への変換器を作成する.本研究の目的は,この変換器を用いて日本語テキストを事前編集し,翻訳し難い日本語を翻訳しやすい日本語に変換することで,機械翻訳の英語母語化を向上させることである.

具体的には,pairset-Wiki,Jpara-small,Jpara-big のデータセットを用いて BART モデルをファインチューニングする.pairset-Wiki データセットは 700 の高品質な日本語と母語話者英語の文ペアを含み,Jpara-small データセットは 2386 の翻訳し難い日本語と母語話者英語の文ペアを含み,Jpara-big データセットは 13000 の日本語と母語話者英語の文ペアを含む.これらのデータセットは,BART モデルが複雑な日本語文を機械翻訳に適した簡単な文に変換する方法を学習するための豊富な訓練データを提供する.

BART (Bidirectional and Auto-Regressive Transformers) は,Facebook AI 研究チームによって提案された,Transformer アーキテクチャに基づくシーケンス・トゥ・シーケンスモデルである.BART は BERT の双方向エンコーダと GPT の自己回帰デコーダを組み合わせ,テキスト生成タスクで優れた性能を発揮する.BART モデルのアーキテクチャにはいくつかの重要な部分が含まれる.まずエンコーダは,BERT に似ており,双方向注意メカニズムを使用して入力テキストをエンコードする.エンコーダはテキストの全体的な文脈情報を捉え,高品質なテキスト表現を生成する.次にデコーダは,GPT に似ており,自己回帰メカニズムを使用してエンコーダの出力に基づいて目標テキストを生成する.デコーダは各単語を生成する際に,以前に生成された単語を考慮し,テキストの一貫性と整合性を保証する.BART は,まず大規模なテキストデータで事前訓練され,ノイズ除去オートエンコーダ方式でテキスト表現を学習する.事前訓練後,モデルは特定のタスクに応じてファインチューニングされ,異なる応用シーンに適応する.

BART モデルをファインチューニングすることで,事前編集を自動的に行う日本語変換器を構築することを目指している.この変換器は,実際の翻訳前に原文を編集し,最終的な翻訳の質と母語化程度を向上させる.pairset-Wiki,Jpara-small,および Jpara-big データセットのファインチューニングにより,BART モデルは翻訳し難い日本語を翻訳しやすい日本語に変換する方法をより良く学習し,機械翻訳の英語母語化を効果的にサポートすることができる.

5.1 BART を用いた変換器モデル

本節では,BARTモデルを使用して日本語変換器モデルを構築する方法を紹介する.BART（Bidirectional and Auto-Regressive Transformers）は強力なsequence-to-sequenceモデルであり,様々な自然言語処理タスクで優れた性能を発揮する[11].BARTモデルを使用して変換器を構築するには,いくつかの重要なステップがある.具体的には,事前訓練モデルの選定,訓練データの準備,モデルのファインチューニングである.図16はBARTファインチューニングの一般的な流れである.

まず,BARTモデルを使用するためには,対象言語の事前訓練モデルを選定する必要がある.本研究では,エンコーダとデコーダの両方が日本語のBART事前訓練モデルを選択した.この事前訓練モデルは,大量の日本語テキストデータで訓練されており,日本語の言語特性と意味情報を効果的に捉えることができ,後続のファインチューニングプロセスに堅実な基盤を提供する.

次に,BARTのファインチューニングには訓練データの準備が必要である.本研究では,日本語と翻訳しやすい日本語の文ペアを使用する.これらの文ペアは,原文の日本語文と目標の日本語文が意味的に一致している必要があり,単に言語構造を調整するだけで,機械翻訳がより正確に処理できるようにする.具体的には,これらのデータセットはpairset-Wiki,Jpara-small,およびJpara-bigから得られ,これらのデータセットは豊富な訓練例を提供し,モデルが翻訳し難い日本語を翻訳しやすい日本語に変換する方法を学習するのに役立つ.具体的なファインチューニングの詳細は本論文の後で説明する.

ファインチューニングの過程では,上記のデータセットに基づくデータでBARTモデルを訓練し,そのパラメータを調整して,特定の変換タスクに適応させる.この方法により,本研究では高効率な日本語変換器モデルを構築し,自動的に事前編集を行い,機械翻訳の精度と流暢性を向上させることを期待している.BARTモデルのファインチューニングを通じて,強力な日本語変換器を構築し,機械翻訳の品質向上をサポートすることができる.本節では,ファインチューニングプロセスの具体的なステップと技術的詳細について説明する.

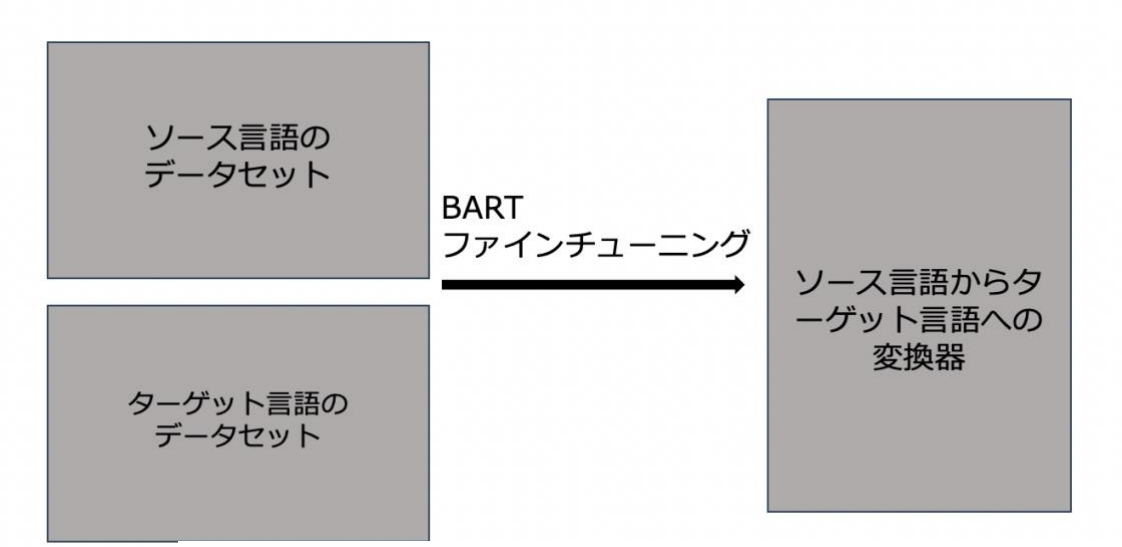

図 16: BART ファインチューニングの一般的な流れ

5.1.1 BART 事前訓練モデル

BART (Bidirectional and Auto-Regressive Transformers) は,Facebook AI チームによって開発された強力なシーケンス・トゥ・シーケンスモデルであり,自然言語生成,翻訳,および理解タスクに使用される.BART は去ノイズオートエンコーダ方式で事前訓練され,多くの自然言語処理タスクで優れた性能を発揮する.BART のアーキテクチャは Transformer モデルに基づいており,双方向エンコーダと自己回帰デコーダを含む.

BART のエンコーダは BERT (Bidirectional Encoder Representations from Transformers)に似ており,双方向注意メカニズムを使用する.双方向エンコーダは,テキストの前向きおよび後向きの文脈情報を同時に捉えることができ,高品質なテキスト表現を生成する.デコーダ部分はGPT[12](Generative Pre-trained Transformer) に似ており,自己回帰メカニズムを使用する.BART と BERT と GPT モデルの関係は図 17 の示すようになる.各単語を生成する際に,前に生成された単語を考慮するため,生成されるテキストの連続性と流暢性が確保される.さらに,デコーダの各層はエンコーダの最終隠れ層に対してクロスアテンションを実行し,エンコーダの出力情報が効果的にデコーダに伝達され,テキスト生成の精度が向上する.

BART の事前訓練プロセスには,テキストの破壊と再構成の 2 つの主要な段階がある.テキスト破壊段階では,任意のノイズ関数を使用してテキストを破壊

する.一般的な方法には,文の順序をランダムに入れ替えることや,テキストの一部を[MASK]記号に置き換えるテキストインフィリングなどが含まれる.テキスト再構成段階では,破壊されたテキストから元のテキストを再構築するシーケンス・トゥ・シーケンスモデルを学習する.再構成損失（交差エントロピー損失）を最適化し,モデルが破壊された入力から元のテキストを再構築できるようにする.

BART は,多様なノイズ関数を使用して入力テキストを破壊する.これには,単語マスク（Token Masking）,単語削除（Token Deletion）,テキストインフィリング（Text Infilling）,文の順序入れ替え（Sentence Permutation）,および文書回転（Document Rotation）が含まれる.これらのノイズ方法により,BART モデルは単語レベルのマスクを学習するだけでなく,テキストの全体的な長さや構造の変換にも対応できる.

BART は,テキスト生成,要約生成,質問応答システム,機械翻訳タスクなどの多くの自然言語処理タスクで優れた性能を発揮する.BART の自己回帰デコーダはテキスト生成タスクで優れており,ニュース記事や文書の要約生成タスクでは新たな性能のピークを達成した.質問応答タスクでは,BART は正確な回答を生成でき,機械翻訳タスクでは追加の Transformer 層を使用することで翻訳品質を向上させた.

BART のファインチューニングプロセスは,特定のタスクのデータセットで訓練し,モデルパラメータを調整して特定のタスクに適応させることを含む.ファインチューニングのステップには,タスクデータの準備,訓練パラメータの設定,およびモデル訓練が含まれる.タスクデータセットでモデルを訓練し,検証セットの性能に基づいて調整を行う.これらのステップを通じて,BART は特定のタスクで優れた性能を発揮し,強力な生成および理解能力を示す.

総じて,BART はその双方向エンコーダと自己回帰デコーダの独自のアーキテクチャ,および柔軟な事前訓練とファインチューニング方法により,自然言語処理タスクで強力な性能を発揮する.BART モデルの詳細な紹介を通じて,その生成,翻訳,および理解タスクにおける広範な応用可能性が明らかになった.

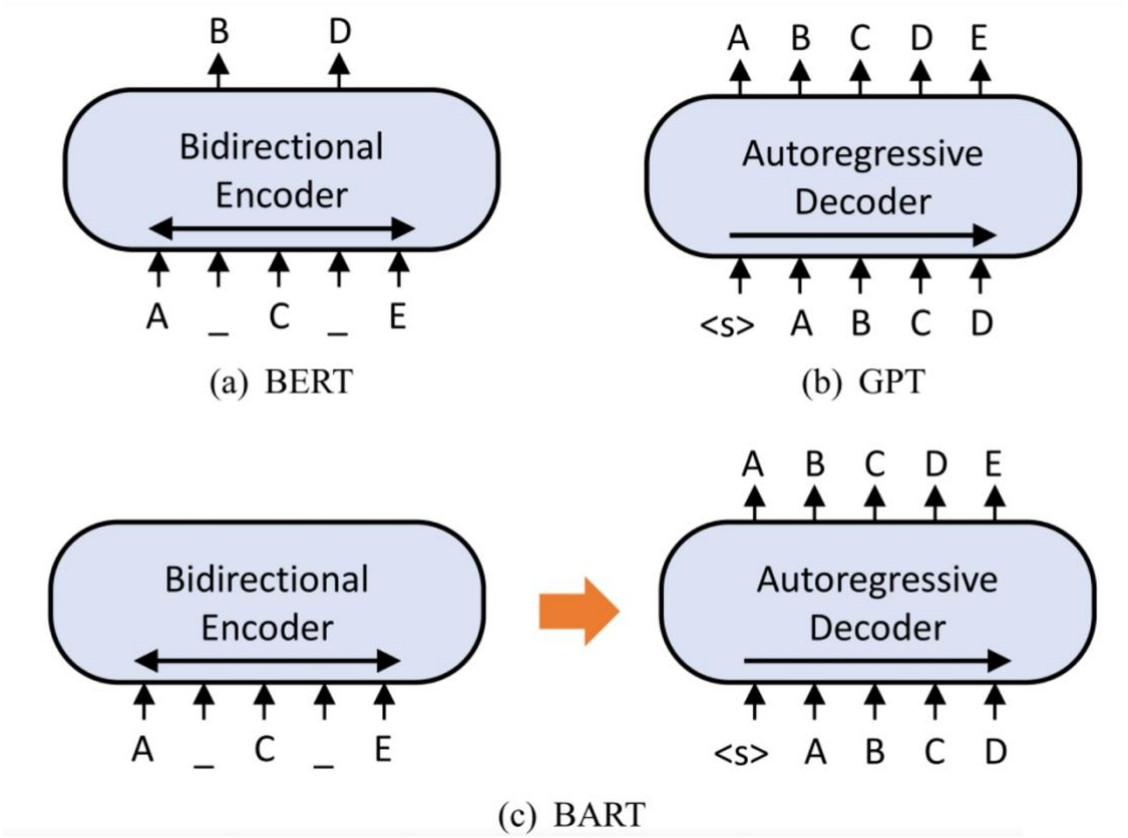

図 17:BART と BERT と GPT モデルの関係

5.1.2 ファインチューニング

本節では,Fairseq で内蔵するBART base 2.0モデルを使用してFairseqフレームワークでファインチューニングを行う方法を詳細に紹介する.BARTモデルは日本語のウィキペディアで事前訓練されており,具体的な事前訓練タスクにはテキストインフィリングタスク（text infilling task）と文の順序並べ替えタスク（sentence permutation task）が含まれる.

BARTモデルのファインチューニングプロセスは,適切な事前訓練モデルの選定から始まる.本研究では,大量の日本語ウィキペディアテキストで事前訓練されたBART base 2.0モデルを選択した.事前訓練タスクには,テキストインフィリングタスクと文の順序並べ替えタスクが含まれる.テキストインフィリングタスクでは,モデルは一部の単語が欠けた文を埋めることで文脈関係を学習する.文の順序並べ替えタスクでは,モデルは順序が乱れた文を再構成して文間の論理的な順序を把握する.

ファインチューニングにはFairseqを使用する.FairseqはFacebook AI Researchによって開発されたシーケンス・トゥ・シーケンス学習のためのフレームワークであり,Transformer,LSTM,畳み込みニューラルネットワークなどの多様なモデルアーキテクチャをサポートしている.Fairseqは高度にモジュール化されており,研究者がモデルを簡単にカスタマイズおよび拡張できる.各モジュールは独立して置き換えや調整が可能で,さまざまな研究ニーズに適応できる.

さらに,Fairseqは分散訓練と混合精度訓練をサポートしており,大規模データセットで効率的にモデルを訓練できる.分散訓練により複数のGPUを同時に使用して訓練時間を大幅に短縮でき,混合精度訓練は訓練中に計算精度を動的に調整し,メモリ使用量を削減して計算効率を向上させる.FairseqにはBART,BERT,RoBERTaなどの多くの事前訓練モデルが内蔵されており,これらのモデルは大量のデータで訓練されて高い汎用性を持ち,研究者はこれを基にファインチューニングを行い,特定のタスクに迅速に適応させることができる.事前訓練モデルは強力なスタート地点を提供し,研究者はモデルを一から訓練する必要がなく,時間と計算リソースを節約できる.

Fairseqはテキスト生成や機械翻訳タスクだけでなく,言語モデル,テキスト分類,質問応答などのさまざまな自然言語処理タスクもサポートしている.研究者は具体的なニーズに応じて適切なタスクとモデルアーキテクチャを選択し,対応する訓練と評価を行うことができる.この多タスクサポートにより,Fairseqは異なる研究方向や応用シーンに適応できる柔軟なツールとなっている.

Fairseqは活発なコミュニティと詳細なドキュメントサポートを持ち,研究者は使用中に公式ドキュメントを参照して関連技術サポートやリソースを得ることができる.さらに,Fairseqのオープンソース特性により,研究者は自身の研究成果を共有し交流することで,技術の共同進歩を促進できる.

ファインチューニングの過程では,まずpairset-Wiki,Jpara-small,Jpara-bigデータセットをFairseqフレームワークが受け入れられる形式にフォーマットする.これらのデータセットは日本語原文と翻訳しやすい日本語のペアを含み,意味の一貫性を確保する.次に,学習率,バッチサイズ,オプティマイザなどの訓練パラメータを設定し,具体的なタスクのニーズに応じてこれらのパラメータを調整し,最良の性能を得る.続いて,Fairseqを使用してモデルを訓練する.訓練中,モデルはパラメータを調整して特定の変換タスクにより適応するようになる.最後に,検証セットでモデルの性能を評価し,評価結果に基づいてモデルパラメー

タをさらに調整し,異なるテストセットでのモデルのパフォーマンスが期待通りであることを確認する.

これらのステップを通じて,BARTモデルは特定のタスクで優れた性能を発揮し,強力な生成および理解能力を示す.Fairseqは豊富なツールと柔軟な構成を提供し,ファインチューニングプロセスを効率的かつ便利にする.BART base 2.0モデルに基づくファインチューニングを通じて,高効率な日本語変換器モデルを構築し,機械翻訳の精度と流暢性を向上させることを期待している.

5.2 訓練データ

本小節では,BARTモデルをファインチューニングするために,前章で得られた3つのデータセットから適切な訓練データをどのように取得するかを詳細に説明する.これらのデータセットは,pairset-Wiki,Jpara-small,およびJpara-bigである.訓練データを構築するために,これら3つのデータベースから得られた文ペアをFairseqが受け入れられる形式にフォーマットし,データセットの多様性と包括性を確保する.

5.2.1 訓練データの作り方

訓練データを作成するために,前述のpairset-Wiki,Jpara-smallおよびJpara-bigという3つのデータセットを処理する必要がある.まず,母語話者英語から翻訳しやすい日本語を作るために,これらのデータセット内の英語テキストをDeepLで日本語に翻訳する.DeepLは高品質な機械翻訳ツールであり,英語テキストを正確に日本語に翻訳し,翻訳テキストの意味の一貫性を保証できる.そして機械翻訳でできた日本語訳文と日本語原文と組み合わせ,日本語と翻訳しやすい日本語のペアを作る.

日本語と翻訳しやすい日本語の文ペアを得た後,これらの文をトークナイズしてBARTモデルが処理できるようにする必要がある.本研究では,JUMAN++を使用して日語のトークナイズを行う.JUMAN++は京都大学によって開発された効率的な日本語形態解析器であり,形態解析だけでなく,品詞タグ付けや基本形の還元も行うことができる.

JUMAN++の高効率な形態解析により,日本語テキストを正確に単語単位に分割し,各単語の品詞と基本形を認識することができる.さらに,JUMAN++は各トークナイズ単位に対して詳細な品詞タグ付けを行うことができる.これには,名詞,

動詞,形容詞,副詞などが含まれ,日本語文の文法構造と意味情報の理解に非常に重要である.トークナイズと品詞タグ付けを行うと同時に,JUMAN++は動詞や形容詞などの単語を基本形に還元することもできる.これにより,異なる形態を持つ単語を統一し,後続の処理を簡便にすることができる.JUMAN++は豊富な辞書を備えており,一般的な単語から専門用語まで多くの日本語単語をカバーしているため,さまざまな応用シーンで優れた性能を発揮する.JUMAN++は多くのインターフェースを提供しており,さまざまな自然言語処理システムに簡単に統合できる.Python,C++などの複数のプログラミング言語をサポートしており,非常に柔軟に使用できる.本研究ではトークナイズ機能のみを使用し,単語の品詞などの特性は使用しない.

これらのステップを通じて,高品質な訓練データを作成することができ,BARTモデルは翻訳し難い日本語を翻訳しやすい日本語に変換する方法をよりよく学習し,機械翻訳の正確性と流暢性を向上させることができる.

5.2.2 ファインチューニング用のデータ

次に,ファインチューニング用のデータについて説明する.前述の訓練データを作る方法で 3 つの日本語と英語母語話者英語の対訳データセットを処理することで三つのファインチューニング用のデータセットを作った.pairset-Wikiから得られたファインチューニングデータは finetuning_data_v1 と呼ぶ.Jpara-small から得られたデータは finetuning_data_v2 と呼び,Jpara-bigから得られたデータは finetuning_data_v3 と呼ぶ.各バージョンのファインチューニングデータと対訳データセットの関係及びデータセットのサイズは表 4の示すようである.ファインチューニング用のデータを作る流れは図 18 の示すようである.

表 4: 対訳データセットとファインチューニング用のデータの関係

対訳データセット	pairset-wiki	Jpara-small	Jpara-big
ファインチューニングデータ	finetuning_data_v1	finetuning_data_v2	finetuning_data_v3
データサイズ	1000 件	2380 件	13000 件
上記のデータで作ったモデル	model_Wiki	model_Jpara_small	model_Jpara_big

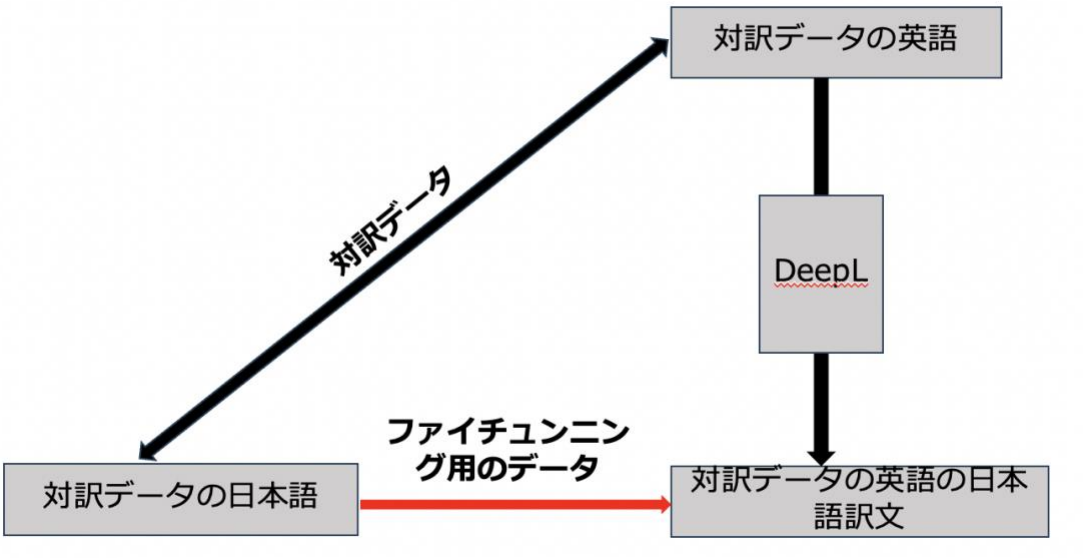

図 18:ファインチューニング用のデータを作る流れ

5.3 日本語から翻訳しやすい日本語への変換器

上述の 3 つのデータセットに基づき,本研究では Fairseq を用いて BART モデルをファインチューニングし,翻訳しやすい日本語への変換器の 3 つのバージョンを作成した.本研究では,training_data_v1 を使用して作成した変換器をモデル v1 と呼び,training_data_v2 を使用して作成した変換器をモデル v2 と呼び,training_data_v3 を使用して作成した変換器をモデル v3 と呼ぶ.次章では,3 つのモデルの性能を詳細に分析する.Fairseq を使用して各モデルを訓練するためのパラメータの詳細は表 5 に示されている.

表 5: 各モデルを訓練する際のパラメータ

parameter	modelv1	model v2 and v3
arch	bart_base	bart_base
restore-file	japanese_bart_base_2.0/bart_model.pt	japanese_bart_base_2.0/bart_model.pt
save-dir	$SAVE_MODEL_DIR	$SAVE_MODEL_DIR
tensorboard-logdir	$TENSORBOARD_DIR	$TENSORBOARD_DIR
task	translation_from_pretrained_bart	translation_from_pretrained_bart
source-lang	src	src
target-lang	tgt	tgt
criterion	label_smoothed_cross_entropy	label_smoothed_cross_entropy
label-smoothing	0.2	0.2
dataset-impl	raw	raw
optimizer	adam	adam
adam-eps	1.00E-06	1.00E-06
adam-betas	'{0.9, 0.98}'	'{0.9, 0.98}'
lr	3.00E-05	3.00E-05
warmup-updates	2500	500
total-num-update	40000	40000
dropout	0.3	0.3
max-tokens	1024	1024
update-freq	2	1
max-update	80000	20000

第6章 モデル評価

この章では,上述の3つの日本語から翻訳しやすい日本語への変換器の性能を評価する.本章では,評価指標と評価データを紹介する.また,本研究の日本語変換器だけではない,ChatGPT-4o を用いて日本語の事前編集を行い,本研究の変換器と ChatGPT が異なるデータセットでの性能を比較する.

6.1 性能指標

本研究の評価指標は,日本語の英訳文の母語化程度である.具体的には,日本語を機械翻訳で英語に翻訳し,その後,英語スタイル分類器を使用して英訳文の母語化程度を評価する.事前編集を行った日本語と事前編集を行っていない日本語の英訳文の英語母語化程度を比較することで,日本語変換モデルの性能を評価することができる.具体的に,各テストデータセットの各文の英語母語化程度の平均値をデータセット全体の英語母語化程度として評価する.同時に,各データセットにおける日本人英語と母語話者英語の割合も集計する.テストデータセット全体の英語母語化程度とその中の母語話者英語の割合をモデルの評価指標とする

6.2 テストコーパス

評価結果の信頼性を確保するために,複数の異なるデータセットを使用してモデルを評価した.テストデータには,ウィキペディア,Jpara 日英コーパス,および livedoor ニュースコーパスからの 3 つのテストデータセットを含む:

1. JA-original-ドメイン固有:京都に関するウィキペディアページからランダムに選ばれた 2558 文の完全な日本語.このデータセットは,百科事典形式のテキストに対するモデルの性能を評価するために使用される.
2. JA-original-汎用 : Jpara コーパスから取得した 240 文の完全な日本語.このデータセットは,対話と翻訳コーパスにおけるモデルの性能を評価するために使用される.
3. JA-original-formal : livedoor ニュースコーパスから選ばれた 3000 文の完全な日本語.このデータセットは,ニュースや報道形式のテキストに対するモデルの性能を評価するために使用される.

これらのテストデータはすべて訓練データには含まれておらず,テストの公平性と信頼性を確保している.

6.3 事前編集の効果検証

6.3.1 JA-original-ドメイン固有を用いた事前編集の効果検証

本研究では,JA-original-ドメイン固有の日本語文を以下のように処理し,これらのデータの評価結果を計算する：

1. JA-original-ドメイン固有の日本語文をDeepLで英語に翻訳し,これらの英語文を「JA-original-ドメイン固有-Deepl」と呼ぶ.
2. JA-original-ドメイン固有の日本語文をmodel_Wikiで事前編集し,DeepLで英語に翻訳して「JA-model_Wiki-Wiki-Deepl」と呼ぶ.
3. JA-original-ドメイン固有の日本語文をmodel_Jpara_smallで事前編集し,DeepLで英語に翻訳して「JA-model_Jpara_small-Wiki-Deepl」と呼ぶ.
4. JA-original-ドメイン固有の日本語文をmodel_Jpara_bigで事前編集し,DeepLで英語に翻訳して「JA-model_Jpara_big-Wiki-Deepl」と呼ぶ.
5. JA-original-ドメイン固有の日本語文をChatGPT-4で事前編集し,DeepLで英語に翻訳して「JA-gpt_4-Wiki-Deepl」と呼ぶ.ChatGPTを事前編集させるプロンプトは「Modify all of the Japanese in this file so that these Japanese words can be translated by machine translation into English from native English speakers. Don't need to show me the sentences but need to put the modified file into a new txt file.」である.
6. JA-original-ドメイン固有の日本語文をGoogle翻訳で英語に翻訳し,これらの英語文を「JA-original-ドメイン固有-Google」と呼ぶ.
7. JA-original-ドメイン固有の日本語文を model_Jpara_big で事前編集し,Google 翻訳で英語に翻訳して「JA-model_Jpara_big-Wiki-Google」と呼ぶ.

Deepl を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合は表 6 の示すようである. 表 6 の示すように,三つの日本語変換器は Wikipedia のデータに対して効果があることはわかった.その中に,model_Jpara_big は一番良い性能を持っている.モデルの汎用性も評価するために Google 翻訳を用いた場合に JA-original-ドメイン固有-Google と JA-model_Jpara_big-Wiki-Google の母語化程度も計算した. 表 7 の示すように日本語の変換モデルは Google 翻訳にも役に立つことができる.

表 6: Deepl を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合

native/japanese	JA-original-ドメイン固有-Deepl (base)	JA-model_Wiki-Wiki-Deepl	JA-model_Jpara_small-Wiki-Deepl	JA-model_Jpara_big-Wiki-Deepl	JA-gpt_4-Wiki-Deepl
英語母語化程度	[0.412,0.588]	[0.425, 0.574]	[0.434,0.565]	[0.447,0.552]	[0.413,0.587]
英語母語化程度の変化		1.30%	2.20%	3.50%	0.10%
英語母語話者英語の割合	40.50%	41.50%	42.40%	43.70%	40.60%
英語母語話者英語の割合の変化		1%	1.90%	3.20%	0.10%

表 7: Google 翻訳を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合

native/japanese	JA-original-ドメイン固有-Google (base)	JA-model_Jpara_big-Wiki-Google
英語母語化程度	[0.417, 0.582]	[0. 453, 0.546]
英語母語化程度の変化		3.60%
母語話者英語の割合	40.30%	43.90%
母語話者英語の割合の変化		3.60%

6.3.2 JA-original-汎用を用いた事前編集の効果検証

本研究では,JA-original-汎用の日本語文を以下のように処理し,これらのデータの評価結果を計算する：

1.JA-original-汎用の日本語文を DeepL で英語に翻訳し,これらの英語文「JA-original-汎用-Deepl」と呼ぶ.

2.JA-original-汎用の日本語文を model_Wiki で事前編集し,DeepL で英語に翻訳して「JA-model_Wiki-Jpara-Deepl」と呼ぶ.

3.JA-original-汎用の日本語文を model_Jpara_small で事前編集し,DeepL で英語に翻訳して「JA-model_Jpara_small-Jpara-Deepl」と呼ぶ.

4.JA-original-汎用の日本語文を model_Jpara_big で事前編集し,DeepL で英語に翻訳して「JA-model_Jpara_big-Jpara-Deepl」と呼ぶ.

5. JA-original-汎用の日本語文を ChatGPT-4 で事前編集し,DeepL で英語に翻訳して「JA-gpt_4-Jpara-Deepl」と呼ぶ.
6.JA-original-汎用の日本語文を Google 翻訳で英語に翻訳し,これらの英語文を「JA-original-汎用-Google」と呼ぶ.
7.JA-original-汎用の日本語文を model_Jpara_big で事前編集し,Google 翻訳で英語に翻訳して「JA-model_Jpara_big-Jpara-Google」と呼ぶ.

Deepl 翻訳を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合は表 8 の示すようである. 表 8 の示すように,三つの日本語変換器は Wikipedia のデータに対して効果があることはわかった.その中に,model_Jpara_big は一番良い性能を持っている.モデルの汎用性も評価するために Google 翻訳を用いた場合に JA-original-汎用-Google と JA-model_Jpara_big-Jpara-Google の母語化程度も計算した.表 9 の示すように,Google 翻訳を使う場合に Jpara からの 240 件のデータに対して変換モデルの効果は良くない.今度の Jpara データは Deepl にとって翻訳し難いデータしかないので Google 翻訳を使う場合にモデルがこういうデータに対する性能が良くないのは偶然の可能性があると考えられる.

表 8: Deepl 翻訳を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合

native/japanese	JA-original-汎用-Deepl (base)	JA-model_Wiki-Jpara-Deepl	JA-model_Jpara_small-Jpara-Deepl	JA-model_Jpara_big-Jpara-Deepl	JA-gpt_4-Jpara-Deepl
英語母語化程度	[0.245, 0.755]	[0. 289, 0.710]	[0.310,0.689]	[0.352,0.648]	[0.290,0.710]
英語母語化程度の変化		4.40%	6.50%	10.70%	4.50%
母語話者英語の割合	4.20%	14.60%	20.40%	26.70%	16.00%
母語話者英語の割合の変化		10.40%	16.20%	22.50%	11.80%

表 9: Google 翻訳を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合

native/japanese	JA-original-汎用-Google (base)	JA-model_Jpara_big-Jpara-Google
英語母語化程度	[0.464, 0.536]	[0. 444, 0.556]
英語母語化程度の変化		-2.00%
母語話者英語の割合	45.40%	43.30%
母語話者英語の割合の変化		-2.10%

6.3.3 JA-original-formal を用いた事前編集の効果検証

本研究では,JA-original-formalの日本語文を以下のように処理し,これらのデータの評価結果を計算する：

1. JA-original-formalの日本語文をDeepLで英語に翻訳し,これらの英語文を「JA-original-formal-Deepl」と呼ぶ.

2. JA-original-formalの日本語文をmodel_Jpara_bigで事前編集し,DeepLで英語に翻訳して「JA-model_Jpara_big-news-Deepl」と呼ぶ.

2. JA-original-formalの日本語文をchatgpt-4oで事前編集し,DeepLで英語に翻訳して「JA-gpt_4-news-Deepl」と呼ぶ.

3. JA-original-formalの日本語文をGoogle翻訳で英語に翻訳し,これらの英語文を「JA-original-formal-Google」と呼ぶ.

4. JA-original-formal の日本語文を model_Jpara_big で事前編集し,Google 翻訳で英語に翻訳して「JA-model_Jpara_big-news-Google」と呼ぶ.

Deepl を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合は表 10 の示すようである. 表 10 の示すように,日本語変換器は日本語ニュースに対して効果があることはわかった.その中に,model_Jpara_big は一番良い性能を持っている.モデルの汎用性も評価するために Google 翻訳を用いた場合に JA-original-formal-Google と JA-model_Jpara_big-news-Google の母語化程度も計算した. 表 11 の示すように日本語の変換モデルは Google 翻訳にも効果がある.

表 10: Deepl 翻訳を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合

native/japanese	JA-original-formal-Deepl (base)	JA-model_Jpara_big-news-Deepl	JA-gpt_4-news-Deepl
英語母語化程度	[0.466, 0.534]	[0.498, 0.502]	[0.454, 0.546]
英語母語化程度の変化		3.20%	-1.20%
母語話者英語の割合	43.90%	48.20%	42.80%
母語話者英語の割合の変化		4.30%	-1.10%

表 11: Deepl 翻訳を用いた場合にそれぞれのデータの母語化程度と英語母語話者英語の割合

native/japanese	JA-original-formal-Google (base)	JA-model_Jpara_big-news-Google
英語母語化程度	[0.527, 0.473]	[0.534, 0.466]
英語母語化程度の変化		0.70%
母語話者英語の割合	53.40%	53.60%
母語話者英語の割合の変化		0.20%

6.4 考察

6.4.1 事前編集の有効性について

以上のデータの検証結果に基づき,日本語から翻訳しやすい日本語への変換器を用いて事前編集を行うことで,機械翻訳の英語母語化程度を向上させることが有効であることがわかった.また,3つのバージョンの日本語変換器の中で,model_Jpara_bigは常に最も優れた性能を示した.これは,model_Jpara_bigが最も多くの訓練データを使用したためである可能性がある.

さらに,本研究ではJA-original-ドメイン固有およびJA-original-formalデータセットにおいて,日本語変換器を使用することでGoogle翻訳の英語英語母語化程度も向上させることができた.つまり,日本語変換モデルはDeeplだけではない,Google翻訳の英語母語化程度を向上できる.これにより,日本語変換器の汎用性が証明された.JA-original-汎用データセットおよびGoogle翻訳を使用した場合に変換器が期待通りの性能を発揮しなかった理由は,データ量の少なさや,元々DeepLで翻訳が難しい日本語文から来ているためかもしれない.

最後に,DeepL を使用する場合において,model_Jpara_big と ChatGPT-4 で事前編集を行った結果を比較したところ,model_Jpara_big がより良い性能を示した.これにより,本研究の有効性と革新性がさらに強調された.

6.4.2 事前編集の役に立つパターン

本小節では,事前編集が日本語テキストの変換パターンに与える影響をまとめる.複数の例を分析した結果,事前編集が日本語テキストを改善し,機械翻訳が容易になる主な方法として以下の三つが挙げられることが分かった:

1. 不完全な日本語文を補足する
2. 翻訳し難い部分を削除する
3. 日本語文をより適切な言い方に変換する

まずは不完全な日本語文を補足するについて説明する.「月々の支払いから海外旅行まで,万全のサポートを。」という日本語文に対して model_Jpara_big で変換後の日本語は「月々の支払いから海外旅行まで,万全のサポートが必要だ。」.原文の英訳は「Full support from monthly payments to international travel.」.英語母語化程度は[0.11,0.88].変換後の日本語の英訳は「We need full support from monthly payments to international travel.」英語母語化程度は[0.64,0.36]である. これは文の構造を補完することで,文がより完全で明確になり,機械翻訳の結果の英語母語化程度が 0.11 から 0.64 に大幅に向上し,翻訳の正確性が大いに高まった.

次に翻訳し難い部分を削除するパターンについて説明する. 「すごい少なくなってるなぁっていう気がしますね。」という日本語文に対して model_Jpara_big で変換後の日本語は「とても少なくなっているなぁっていう気がする。」.原文の英訳は「I feel that the number is decreasing.」.英語母語化程度は[0.30,0.70].変換後の日本語の英訳は「I feel that there are fewer and fewer of them.」英語母語化程度は[0.57,0.43]である. 翻訳が難しい部分を削除することで,文がより簡潔でわかりやすくなり,翻訳結果の英語母語化程度が 0.303 から 0.571 に向上し,翻訳の質が明らかに改善された.

最後は日本語文をより適切な言い方に変換するについて説明する. 「本作は,鑑賞者が花の中に埋没し庭と一体化する庭園である。」という日本語文に対して model_Jpara_big で変換後の日本語は「この庭園は,鑑賞者が花の中に埋没し,庭と一体化する。」.原文の英訳は「This work is a garden in which the viewer becomes one with the garden by being immersed in the flowers.」.英語母語化程度は

[0.39, 0.61]. 変換後の日本語の英訳は「In this garden, the viewer burrows into the flowers and becomes one with the garden.」英語母語化程度は[0.94, 0.06]である. 文の言い方を変更することで, 文の構造がより自然になり, 翻訳結果の英語母語化度が 0.389 から 0.943 に向上し, 翻訳結果がより流暢でターゲット言語の習慣に適合するようになった.

6.4.3 各データセットに対する性能が違う原因分析

モデルが異なるデータセットで異なる性能を示す原因は複数存在する. まず, テキストのスタイルおよびテーマの違いが重要な要因である. JA-original-汎用データセットには, 歴史, 自然, 科学など多様な内容が含まれ, 文体は比較的正式であり, 多くが陳述文である. それに対して, JA-original-formal データセットは, 主にニュースや時事報道に関連し, 会話体や簡潔なニュース文体が多く見られる. このような文体の違いは, モデルが特定の表現形式を処理する際に影響を及ぼし, その結果, 性能差異が生じる可能性がある.

次に, 語彙およびテーマ領域の違いもモデルの性能差異の重要な要因である. JA-original-汎用では, 専門用語や歴史的な出来事に関連する語彙が多く含まれる一方で, JA-original-formal には時事的な話題や日常的な表現が多く見られる. モデルが訓練時に特定の領域の語彙分布に対して感度が高い場合, 異なる領域のデータセットでは性能が低下することが予測される. この語彙やテーマの違いにより, 特定領域におけるモデルの優位性が, 他の領域では発揮されない可能性がある.

最後に, モデルの過学習も考慮すべき問題である. モデルが訓練時に JA-original-汎用のようなデータセットの文体に過度に適応してしまうと, 異なる文体を持つ JA-original-formal のようなデータに対しては一般化性能が弱くなる可能性がある. このため, モデルは特定のテキストスタイルに依存する傾向があり, 異なるスタイルのテキストに対して同様の性能を示すことが難しくなる.

第7章 おわりに

本研究では,日本語から英語への翻訳において,英語母語話者の英語を生成するための機械翻訳の事前編集手法を提案した. 事前編集とは,機械翻訳で翻訳する前に原文を機械翻訳に適した形に変換することである. 特に,本研究では,現状の機械翻訳を通して英語母語話者の英語を生成しやすい日本語文を「翻訳しやすい日本語」と呼び,原文を翻訳しやすい日本語に変換することを目的とする. 具体的には,日英の対訳コーパスから日本語文と翻訳しやすい日本語のペアを抽出し,このデータで BART をファインチューニングすることで,日本語文から翻訳しやすい日本語への変換器を構築した.

本研究の貢献は以下の 2 点である.

日本語文-翻訳しやすい日本語文のペアを取得した

日本語文から翻訳しやすい日本語への日本語変換器を構築するために,日本語文と翻訳しやすい日本語文のペアが必要である.日本語文は変換前の翻訳しにくい日本語でなければならない.翻訳しにくい日本語は機械翻訳によって日本人英語が生成される日本語文であるため,日本語文を一度機械翻訳によって翻訳しなければならず,大規模なデータを構築するには,翻訳コストが非常に高い. そこで,翻訳コストを抑えるために,できる限り翻訳しにくい日本語と推定されるものに原文段階で限定しなければならない.本研究では,Wikipedia と小説コーパスと Jpara から三つの日本語文と翻訳しやすい日本語文のデータセットを作った.

日本語文から翻訳しやすい日本語への日本語変換器を構築した

日本語文と翻訳しやすい日本語文のペアを用いて日本語変換器を構築した.三つのテストデータの検証結果に基づき,日本語から翻訳しやすい日本語への変換器を用いて事前編集を行うことで,機械翻訳の英語母語化程度を向上させることが有効であることがわかった.また,3 つのバージョンの日本語変換器の中で,model_Jpara_big は常に最も優れた性能を示した. 最後に,DeepL を使用する場合において,model_Jpara_big と ChatGPT-4 で事前編集を行った結果を比較したところ,model_Jpara_big がより良い性能を示した.

本研究では,訓練データ量が多いモデルほど優れた性能を示したことから,将来的にはさらに多くのデータを収集し,モデルの事前編集能力を一層向上させ

ることができると示唆された. さらに, 機械翻訳の英語母語化程度を向上させる方法は他にも存在する. 例えば, 英語のテキスト翻訳モデルをファインチューニングし, 機械翻訳の英語文を事後編集する方法がある. 本研究のさらなる改良やより革新的な研究を期待している.

謝辞

本研究を行うにあたり,熱心なご指導,ご助言を賜りました指導教官の村上陽平教授と Mondheera Pituxcoosuvarn 助教に深謝申し上げます.また,この四年間普段からお世話になっていた社会知能研究室の皆さまに心より感謝申し上げます.

参考文献

[1] Štajner, Sanja, and Maja Popović. "Can text simplification help machine translation?." Proceedings of the 19th Annual Conference of the European Association for Machine Translation. (2016).

[2] Killman, Jeffrey, and Mónica Rodríguez-Castro. "POST-EDITING VS. TRANSLATING IN THE LEGAL CONTEXT: QUALITY AND TIME EFFECTS FROM ENGLISH TO SPANISH." Journal of Language & Law/Revista de Llengua i Dret 78 (2022).

[3] De Almeida, Giselle, and Sharon O'Brien. "Analysing post-editing performance: correlations with years of translation experience." Proceedings of the 14th annual conference of the European association for machine translation. (2010).

[4] Mercader-Alarcón, Julia, and Felipe Sánchez-Martínez. "Analysis of translation errors and evaluation of pre-editing rules for the translation of English news texts into Spanish with Lucy LT." (2016).

[5] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[6] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[7] Targ, Sasha, Diogo Almeida, and Kevin Lyman. "Resnet in resnet: Generalizing residual architectures." arXiv preprint arXiv:1603.08029 (2016).

[8] Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).

[9] Morishita, Makoto, Jun Suzuki, and Masaaki Nagata. "JParaCrawl: A large scale web-based English-Japanese parallel corpus." *arXiv preprint arXiv:1911.10668* (2019).

[10] Sánchez-Cartagena, Víctor, et al. "Prompsit's submission to WMT 2018 parallel corpus filtering shared task." *Proceedings of the third conference on machine translation: shared task papers*. (2018).

[11] Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461* (2019).

[12] Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 pp. 1877-1901(2020).