修士論文

訳語選択の曖昧性を考慮したニューラル機械翻訳

高田久遠

主指導教員 白井 清昭

北陸先端科学技術大学院大学
先端科学技術研究科
（情報科学）

令和８年３月

Abstract

In recent years, the performance of Machine Translation (MT) has improved dramatically due to methods based on neural networks. However, regarding the translation of words with multiple meanings (polysemous words), challenges remain because polysemous words need to be translated into different words depending on the meaning. Word Sense Disambiguation (WSD) plays an important role in translating polysemous words. WSD is the task of identifying the meaning (sense) of a polysemous word appearing in a certain context, and it becomes possible to translate a polysemous word into an appropriate word in the target language by using this as a preprocessing step for translation. On the other hand, since current Neural Machine Translation (NMT) is a method that trains End-to-End models, incorporating WSD as preprocessing has been disregarded. To improve translation performance between Korean and Vietnamese, one of the previous studies identifies the senses of polysemous words using WSD, represents them with different tokens for each sense, and provides sentences with identified word senses as input. However, sense-level tokenization increases the vocabulary size handled by the translation model and increases training parameters, leading to the problem of requiring more training data.

The goal of this study is to explore a method to improve machine translation performance by applying WSD only to translation-oriented ambiguous words and assigning unique tokens for each sense. A translation-oriented ambiguous word is defined as a polysemous word where each sense is translated into a different word in the target language. For example, the English word "bank" has two senses, "financial institution" and "land along a river," which are translated into Japanese as "*ginko*" and "*dote*," respectively; thus, it is a translation-oriented ambiguous word. On the other hand, the word "wall" has two senses, "partition" and "social barrier," but both are translated as "*kabe*" in Japanese, so it is not a translation-oriented ambiguous word. By limiting this sense-level tokenization to translation-oriented ambiguous words, we aim to improve machine translation performance for polysemous words while suppressing the increase in vocabulary size. Note that the languages handled in this study are Japanese as the source language and English as the target language.

First, translation-oriented ambiguous words are automatically identified. A Japanese-English parallel corpus is prepared, and then the word alignment tool GIZA++ is applied for it to identify the correspondence between Japanese words and English words. Next, by aggregating word correspondences across the entire corpus, each of a Japanese word is associated with a list of its multiple corresponding English words. Finally, post-processing such as removing words with low and extremely high frequency is performed. Through this procedure, a lexicon

of translation-oriented ambiguous words is constructed. This lexicon consists of pairs of translation-oriented ambiguous words (Japanese words) and their corresponding multiple English words.

Next, WSD is performed on translation-oriented ambiguous words in the source language sentences of the parallel corpus. The original words are subdivided by sense IDs identified by WSD (e.g., word_senseID) and treated as individual tokens for each sense. After this processing is performed, the MT model is trained using the parallel corpus as training data. The Transformer is employed as the translation model. It is trained from scratch with randomly initialized parameters.

This study proposes multiple methods based on the definition of senses and the WSD method. The first method performs WSD using the Iwanami Japanese Dictionary as the definition of senses. This method is denoted as "JDIC." An existing WSD tool, KyWSD, is used for WSD. This method has the advantage that WSD can be applied to all translation-oriented ambiguous words, but it also has the disadvantage that dictionary senses are not necessarily associated with translated words; that is, different senses in the dictionary are not necessarily translated into different English words. The second method performs WSD using the list of translated words in the lexicon of translation-oriented ambiguous words as the definition of senses. This method is denoted as "TR." However, the WSD model used in TR needs to be trained independently. A dataset to train the WSD model is constructed by labeling Japanese words with their corresponding translations (senses) based on word alignment results from GIZA++. Three WSD models are proposed: (1) a model that uses the pre-trained language model BERT to obtain embeddings of target words and learns multiple Fully Connected Layers (FCLs) to classify the sense of each word using these embeddings as features (called TR-pre), (2) a model that fine-tunes BERT individually for each word (called TR-ft-indi), and (3) a model that learns a BERT model shared among all words and multiple FCLs to classify the sense of each word (called TR-ft-shared). For the translation experiments, we adopted the first and third methods in terms of computational cost and performance. As extensions of the third model, we also train a model that classifies the sense as "unknown" when it does not correspond to any translation in the lexicon of translation-oriented ambiguous words (called TR-ft-shared-unk), and a model that classifies the sense as "uncertain" when the WSD model cannot predict the sense with high confidence (called TR-ft-shared-uncer).

Several experiments are conducted to evaluate the effectiveness of the proposed methods. One million sentence pairs randomly extracted from the Japanese-English parallel corpus JParaCrawl is used as training data, and another 4,000 sentences is used as test data. As evaluation metrics, in addition to BLEU, a

standard metric for MT, a unique metric called "translation selection accuracy" is used. This metric approximately estimates the ratio at which translation-oriented ambiguous words were translated into correct words. The method that trains the translation model without WSD (called vanilla) achieved a BLEU of 27.9 and a translation selection accuracy of 0.523, while the method that performs WSD on all words corresponding to previous research (called JDIC-all) achieved a BLEU of 27.2 and a translation selection accuracy of 0.523. In contrast, among the proposed methods that perform WSD only on translation-oriented ambiguous words, TR-ft-shared achieved the best performance, with a BLEU of 28.1 and a translation selection accuracy of 0.534. JDIC, which uses a dictionary as the definition of senses, did not outperform the baselines (vanilla and JDIC-all). This may be because the dictionary senses are not directly associated with the ambiguity of translation selection. Other proposed methods improved BLEU or translation selection accuracy compared to the baselines, but there were few cases where both were improved, thus their effectiveness was limited compared to TR-ft-shared. Furthermore, when measuring the translation performance of the proposed method under the ideal condition where tokens were subdivided by correct senses, that is, under the situation that the WSD accuracy was 100%, the BLEU became 28.9 and the translation selection accuracy became 0.689. Compared to the vanilla and JDIC-all baselines, the translation selection accuracy improved significantly in particular. This supports the validity of the approach of the proposed method, which performs WSD limited to translation-oriented ambiguous words, subdivides tokens for each sense, and then trains the translation model.

概要

近年、ニューラルネットワークに基づく手法により機械翻訳の性能は飛躍的に向上した。しかし、複数の意味を持つ単語 (多義語) の翻訳については、多義語は意味に応じて異なる単語に訳し分ける必要があるため、依然として課題が残されている。多義語の翻訳において重要な役割を果たすのは語義曖昧性解消 (Word Sense Disambiguation; WSD) である。WSD はある文脈に出現する多義語の意味 (語義) を特定するタスクであり、これを翻訳の前処理とすることで多義語を目標言語の適切な単語に翻訳できるようになる。一方、現在主流のニューラル機械翻訳は End-to-End のモデルを学習する手法であるため、WSD を前処理として組み込むことは軽視されていた。その数少ない研究のひとつとして、韓国語とベトナム語の翻訳を対象に、まず WSD によって多義語の語義を特定し、多義語を語義毎に別のトークンで表現することによって、つまり語義が特定された文を入力として与えることによって翻訳の性能を向上させる手法が提案されている。しかし、語義毎にトークンを分けると翻訳モデルが扱う語彙サイズが増大し、訓練パラメタも増大するため、より多くの訓練データを必要とするという問題点がある。

本研究では、訳語選択曖昧語のみに WSD を適用し、語義毎にトークンを分けることによって機械翻訳の性能を向上させる手法を探究することを目的とする。訳語選択曖昧語とは、複数の語義を持つ多義語のうち、それぞれの語義が目標言語の別の単語に翻訳される単語と定義する。例えば、英語の bank という単語は (金融機関) と (川の周辺の土地) の 2 つの語義を持ち、それぞれ日本語では「銀行」もしくは「土手」と訳されるので、訳語選択曖昧語である。一方、wall という単語には (仕切り) と (社会的障壁) の 2 つの語義があるが、日本語ではともに「壁」と訳されるので、訳語選択曖昧語ではない。トークン分割の対象を訳語選択曖昧語に限定することで、語彙サイズの増加を抑えつつ、多義語に対する機械翻訳の性能を向上させることを狙う。なお、本研究が扱う翻訳の言語は、原言語を日本語、目標言語を英語とする。

まず、訳語選択曖昧語を特定する。日英パラレルコーパスを用意し、単語アライメントツール GIZA++ を使って日本語単語と英単語の対応関係を特定する。次に、コーパス全体における単語の対応関係を集約し、ひとつの日本語単語に対し、それに対応する複数の英単語のリストを対応付ける。最後に、出現頻度が低い単語や極端に高い単語を除去するなどの後処理を行う。以上の手続きで訳語選択曖昧語の辞書が構築される。この辞書は訳語選択曖昧語 (日本語単語) とその対訳となる複数の英単語の組から構成されている。

次に、パラレルコーパスの原言語の文に含まれる訳語選択曖昧語に対し WSD を行う。元の単語を「word_senseID」のように WSD によって特定した語義の ID で細分化し、語義毎に個別のトークンとして扱う。この処理を行った後のパラレルコーパスを訓練データとして翻訳モデルを機械学習する。翻訳モデルとしては Transformer を採用し、初期パラメタをランダムに決めて、スクラッチから学習する。

本研究ではWSDにおける語義の定義ならびにWSDの手法によって複数の手法を提案する。第1の手法は、岩波国語辞典を語義の定義としてWSDを行う。この手法を「JDIC」と記す。WSDには既存のWSDツールであるKyWSDを用いる。この手法は全ての単語に対してWSDを実施できるという利点があるが、辞書の語義は必ずしも翻訳の訳語と対応付けられていない、つまり辞書における異なる語義が異なる英単語に翻訳されるわけではないという欠点もある。第2の手法は、訳語選択曖昧語辞書における訳語のリストを語義の定義としWSDを行う。この手法を「TR」と記す。ただし、TRで用いるWSDモデルは独自に学習する必要がある。GIZA++による単語のアライメント結果を利用し、日本語単語に対してそれに対応する訳語 (語義) をラベル付けしたデータセットを自動構築し、WSDモデルの訓練データとする。また、WSDのモデルとして、(1) 事前学習済み言語モデルBERTを用いて対象単語の埋め込みを取得し、これを特徴量として各単語の語義を分類する複数の全結合層を学習するモデル (TR-pre)、(2) 単語毎に個別にBERTをファインチューニングするモデル (TR-ft-indi)、(3) 全単語に共通のBERTモデルと各単語の語義を分類する複数の全結合層を学習するモデル (TR-ft-shared) の3つを提案する。翻訳実験では、計算コストと性能の観点から1番目と3番目のモデルを採用する。 3番目のモデルの拡張モデルとして、語義が訳語選択曖昧語辞書における訳語のいずれにも該当しないときに「unknown」というクラスに分類するモデル (TR-ft-shared-unk) と、WSDモデルが高い確信度で語義を推定できないときに「uncertain」というクラスに分類するモデル (TR-ft-shared-uncer) も学習する。

提案手法の有効性を評価する実験を行う。日英パラレルコーパスJParaCrawlからランダム抽出した100万文対を訓練データとし、別の4000文をテストデータとする。評価指標としては、機械翻訳の標準的な指標であるBLEUに加え、「訳語選択正解率」という独自の指標を用いる。これは訳語選択曖昧語が正しい訳語に翻訳できた割合を近似的に推定したものである。WSDを行わずに翻訳モデルを学習する手法 (vanilla) のBLEUは27.9、訳語選択正解率は0.523であり、先行研究に該当する全ての単語についてWSDを行う手法 (JDIC-all) のBLEUは27.2、訳語選択正解率は0.523であった。これに対し、訳語選択曖昧語のみにWSDを行う提案手法で最も成績が良かったのはTR-ft-sharedであり、BLEUは28.1、訳語選択正解率は0.534であった。辞書を語義の定義とするJDICはベースライン (vanilla、JDIC-all) を上回ることはなかった。これは辞書の語義が翻訳における訳語選択の曖昧性と直接的に関連付けられていないことが原因と思われる。その他の提案手法は、ベースラインと比べてBLEUや訳語選択選択率が向上したが、両方とも改善されたケースはほとんどなく、その有効性はTR-ft-sharedと比べて限定的であった。また、正解の語義によってトークンを細分化したとき、つまりWSDの正解率が100%という理想的な条件で提案手法の翻訳の性能を測ったところ、BLEUは28.9、訳語選択正解率は0.689となった。ベースラインであるvanillaやJDIC-allと比べて特に訳語選択正解率が大きく改善している。このことは、訳語選択曖昧

語に限って WSD を行い、語義毎にトークンを細分化した上で翻訳モデルを学習するという提案手法のアプローチが妥当であることを示唆している。

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

機械翻訳は、ある言語の文や文章を別の言語に自動的に翻訳するタスクである。機械翻訳は長い間自然言語処理研究分野における中心的な研究課題であった。近年では、ニューラルネットワークに基づく手法により、機械翻訳の性能は飛躍的に向上した。しかし、文脈に応じて意味が変化する多義語の翻訳については、依然として課題が残されている。一般に、単語は複数の意味を持ち、そのような複数の意味を持つ単語は多義語と呼ばれている。一般に多義語ではそれぞれの意味に対する訳語は異なる。例えば、英語の bank という単語には (金融機関) と (川の周辺の土地) という 2 つの意味があり、それぞれの意味は日本語では「銀行」もしくは「土手」と訳される。多義語の翻訳において、適切な訳語を選択するためには、文脈から単語の正しい意味を特定するプロセスが不可欠である。

多義語の翻訳において重要な役割を担うのが語義曖昧性解消 (Word Sense Disambiguation; WSD) である。WSD とは単語の意味 (語義) を特定するタスクである。先ほど述べたように、一般に原言語の単語はその意味に応じて異なる単語に翻訳されるため、WSD は翻訳精度の向上に貢献すると考えられる。

一方で、現在のニューラル機械翻訳 (Neural Machine Translation; NMT) においては、WSD は必ずしも重要視されていない。これは、NMT モデルが文全体の情報を抽象表現 (ベクトル表現) に変換する過程で、文脈に応じた意味の識別が暗黙的に行われていると考えられているためである [1]。これに対し、翻訳前に明示的に WSD を行い、個々の語義を個別のトークンとして扱うことで NMT の性能を向上させた研究もある [2]。しかし、この手法には、翻訳モデルが扱うトークンの数が増えるために、より多くの訓練データを要するといった問題がある。

1.2　目的

本研究は、訳語選択の曖昧性に着目してニューラル機械翻訳の性能を向上させることを目的とする。訳語選択の曖昧性とは、原言語の単語における複数の語義が目標言語で異なる単語に翻訳されることを指す。例えば、英語の bank には 2 つの意味があり、日本語ではそれぞれ「銀行」「土手」のように異なる単語に翻訳されるため、訳語選択の曖昧性がある。一方、wall には (仕切り) と (社会的障壁) の

2 つの意味があるが、日本語ではともに「壁」と翻訳されるため、訳語選択の曖昧性がない。本研究では、目標言語において異なる単語に翻訳される多義語を「訳語選択曖昧語」と定義する。訳語選択の曖昧性がある単語に限って WSD を行い、語義毎にトークンを分けることで、全体のトークン数を抑えつつ、単語を適切に訳し分けることを狙う。

1.3 本論文の構成

本論文の構成は以下の通りである。

2 章では、関連研究と本研究の特徴について述べる。3 章では、訳語選択の曖昧性に着目し機械翻訳の性能を向上させるための提案手法の詳細について述べる。4 章では、提案手法の評価実験、結果、考察について述べる。最後に、5 章では、本論文のまとめと今後の課題について述べる。

第2章　関連研究

2.1　ニューラル機械翻訳における多義語の翻訳の課題

ニューラル機械翻訳（NMT）は、Encoder-Decoder 構造や Attention 機構を用いることで、文全体または局所的な文脈情報を考慮した翻訳が可能である [3, 4, 5]。これにより、従来の統計的機械翻訳（Statical Machine Translation; SMT）と比較して、翻訳の精度が飛躍的に向上した。これに伴い、多義語の訳し分け能力、すなわち複数の語義を持つ単語を語義に応じて適切な訳語に翻訳する能力も改善されている。

しかし、NMT モデルは依然として多義語の翻訳において誤りを犯すことが報告されている。特に、学習データ内で出現頻度の高い語義（Most Frequent Sense; MFS）への翻訳を過剰に優先してしまう傾向がある。Rios らは、大規模な対照テストセット（Contrastive Test Set）を用いて NMT の WSD 能力を評価し、文脈を考慮すると低頻度の語義に対応する訳語が適切である場合でも、NMT が文脈を無視して高頻度語義を出力するケースがあることを指摘している [1]。この実験結果から、NMT モデルは文脈情報を十分に活用できておらず、学習データにおける単語の共起統計や頻度バイアスを優先して翻訳することがあると考察している。

さらに、文脈を利用して曖昧性を解消しようとする試みにおいても課題が残されている。Rippeth らは、長い文脈を持つ文書レベルのデータセット (パラレルコーパス) の収集が困難であるという問題に加え、単に長い文脈を入力するだけでは翻訳精度が下がることを指摘している [6]。そして、類似した文脈を持つ文を集め、疑似文書を作成し、その中からトピックを表す重要単語を明示的に抽出し、それをヒントとして翻訳する原文の先頭に付与することで、文脈情報を補完し、WSD の精度を向上させた。

また、近年急速に発展している大規模言語モデル (Large Language Model; LLM) についても、WSD の能力に限界があること指摘されている。Basile らは、LLM のゼロショットによる WSD 能力を評価した結果、LLM は一定の性能を示すものの、依然として既存の WSD に特化した手法やファインチューニングされた小〜中規模モデルには及ばないケースがあることを報告している [7]。これは、LLM が台頭する現在においても、多義語の厳密な訳し分けに特化した WSD が依然として重要であることを示唆している。

2.2 外部知識を用いたニューラル機械翻訳

NMT は高い翻訳性能を誇る一方で、その性能を発揮するためには大量のパラレルコーパスを必要とする。Koehn と Knowles は、学習データ量が減少すると NMT の性能が急激に劣化することを示し、低資源環境における 6 つの NMT の課題（Six Challenges）の一つとして「低頻度語の翻訳精度」を挙げている [8]。

低資源環境における NMT モデルの性能を向上させるために、外部知識を用いて語義に関する情報を NMT モデルに組み込む手法がいくつか提案されている。これらは知識の与え方によっていくつかの種類に分類できる。

2.2.1 語義埋め込みを利用する手法

Rios らは、語義とその定義文から構築した語義埋め込みや、文章内で意味が類似した単語を繋げた語彙連鎖（Lexical Chains）の埋め込みを NMT モデルへの入力として与えることで、特に低頻度語における WSD 精度を向上させている [1]。

Pu らは、複数の意味を持つ名詞と動詞を対象にクラスタリングベースの WSD 手法を提案し、これに基づいて学習された語義埋め込みを NMT モデルへの入力として与えることで翻訳性能を向上させている [9]。

2.2.2 知識グラフや辞書情報による訳語選択の補助

単純な埋め込みだけでなく、構造化された知識を利用する研究も行われている。Zhao らは、知識グラフ上の語義関係を NMT モデルに取り込むことで、学習データ量が少ない固有表現や多義語の翻訳精度を改善できることを示している [10]。

また、外部辞書の訳語候補を提示し、動的に選択させるアプローチもある。Maheshwari らは、辞書にある複数の訳語候補をすべて提示し、その中から文脈に合うものを翻訳モデル自身に選択させる手法を提案している [11]。訳語候補を絞り込まず辞書を活用して全ての訳語候補を明示的にモデルに与えることで、不自然な翻訳や誤訳を防ぐことができる。また、辞書内の訳語候補の中に適切なものがないと判断した場合、辞書を無視して自力で翻訳するという柔軟な対応も可能にしている。

一方で、正しい訳語候補を提示するのではなく、文脈に合わない誤った訳語の出力を禁止する負の語彙制約（Negative Lexical Constraints）と呼ばれるアプローチがある [12, 13]。しかし、モデルがその単語の別の活用形や派生形を生成してしまい、結果的に制約をすり抜けて訳出されるという問題がある。Jon らは負の語彙制約に基づく手法の改良を探求している [14]。テキスト生成プロセスの改善や学習データを調整する手法など複数の手法を比較し、さらに単語の語幹レベルで制約をかけることで、禁止された単語があらゆる表面的な変化形として出現するのを防ぐ手法を提案している。これにより、負の語彙制約に違反して単語が訳出され

る問題はある程度解消されたが、多くのケースで制約のすり抜けが依然として発生しており、NMT モデルにおいて特定の単語の生成を完全に排除することが困難であることを示している。

2.2.3 語義曖昧性解消を前処理とするニューラル機械

機械翻訳モデルに入力する前に、前処理として WSD を行い、入力トークンを語義毎に細分化して多義語の訳し分けの性能を向上させるアプローチもある。Nguyen らは低資源である韓国語—ベトナム語間の翻訳において、前処理として原言語（韓国語）の文に WSD を行い、語義ごとに個別のトークン ID を与えることで翻訳精度を向上させた [2]。また、類似した単語をグループ化する語彙ネットワークを構築することで、辞書にない単語の語義推測も可能にしている。この手法は、翻訳モデルが暗黙的に多義語の語義を推定する処理を排除し、前に語義を推定して翻訳モデルに与えるものであり、多義語の翻訳の性能を向上させる、最も直接的な解決策の一つである。

トークン自体を細分化するのではなく、WSD により特定された語義を制約やプロンプトとして翻訳プロセスに統合する手法も近年注目されている。特定の単語をユーザーや辞書の指定通りに翻訳させる技術は、語彙制約付き機械翻訳 (Lexically Constrained NMT; LCNMT) と呼ばれ、広く研究されている。しかし、従来の LCNMT では、多義語のように複数の訳語候補が存在する場合、翻訳時にどの訳語を選択すべきかという曖昧性解消の課題があった。

Zhang らは、事前に WSD を行って文脈に最適な 1 つの訳語を特定し、それを NMT に統合する D-LCNMT を提案した [15]。これにより、指定した訳語が正しく出力される確率、ならびに文全体の翻訳指標である BLEU や COMET スコアが向上したと報告している。Tran らは LLM を用いた機械翻訳において、翻訳を行う前に LLM や原言語の WSD モデルを用いて多義語の WSD を行い、予測された 1 つの語義をプロンプトとして組み込む手法を提案し、評価実験によって BLEU や chrF スコアといった機械翻訳の指標が向上したと報告している [16]。

これらの研究は、NMT や LLM の内部処理に直接多義語の曖昧性の解消を委ねるのではなく、前処理として WSD を挟み、語義を 1 つに絞ってから機械翻訳モデルに渡すアプローチが堅牢かつ有効であることを裏付けている。

2.2.4 マルチタスク学習による言語知識の統合

前処理として WSD や言語処理を行う手法に対し、翻訳タスクと他の言語解析タスクを単一のネットワーク内で同時に解くマルチタスク学習のアプローチも提案されている。例えば、Niehues と Cho は、機械翻訳タスクと同時に品詞タグ付けや構文解析などのタスクの訓練データを用いて翻訳モデルを学習させることで、モデルに言語的知識を組み込むマルチタスク学習手法を提案した [17]。しかし、こ

のようなマルチタスク学習はモデル構造が複雑になりやすく、複数のタスクの学習損失のバランスの調整が難しいという課題がある。

2.3 本研究の特徴

本研究では、Nguyen ら [2] の研究に従い、事前に WSD を行って多義語の語義を特定し、異なる語義を個別のトークンとして扱うことで多義語を正しく訳し分けるアプローチを探究する。ただし、全ての単語に対して語義毎にトークンを分けるのではなく、目標言語において別の単語に訳し分けられる単語、すなわち訳語選択曖昧語のみを対象に WSD を行い、語義毎にトークンを分ける。これにより NMT システムが取り扱うべき語彙の量を減らすことで、訓練データ量の不足を補いつつ、翻訳の性能を向上させることを狙う。

第3章　提案手法

本章では提案手法の詳細について述べる。なお、本研究では日英機械翻訳を対象とする。

3.1　概要

提案手法の概要を図3.1に示す。提案手法の手順は大きく分けて3段階に分けられる。

1. **訳語選択曖昧語辞書の構築**

   日英パラレルコーパスに対して、単語アライメントツールGIZA++[18]を用いて単語アライメントを行う。単語アライメントとは、原言語の単語が目標言語のどの単語に翻訳されているかを対応付けるタスクであり、原言語の単語が目標言語の単語に翻訳される確率を推定する処理を含む。GIZA++の出力結果に基づき訳語選択曖昧語を選定し、訳語選択曖昧語辞書を構築する。

2. **WSD モデルの学習**

   訳語選択曖昧語に対し、それに対応する目標言語の単語を語義と定義し、語義の曖昧性を解消するモデル (WSD モデル) を学習する。WSD モデルの訓練データは、パラレルコーパスにおける単語アライメントの結果を基に、原言語の文における対象単語の語義 (対応する目標言語の単語) を自動的にラベル付けして構築する。

3. **NMT モデルの学習**

   WSD モデルで訳語選択曖昧語の語義を識別し、語義毎にトークンを細分化する。以下の例では「正しい」というトークンの語義を推定し、その語義ID(02) をトークンに付与することでトークンを細分化している。

   例、太郎が正しい。 → 太郎 が 正しい_02 。

   WSD によって語義毎にトークンを細分化したパラレルコーパスを訓練データとし、Transformer[19] をベースとしてNMT モデルをスクラッチから学習する。

それぞれのステップの詳細を次節から説明する。

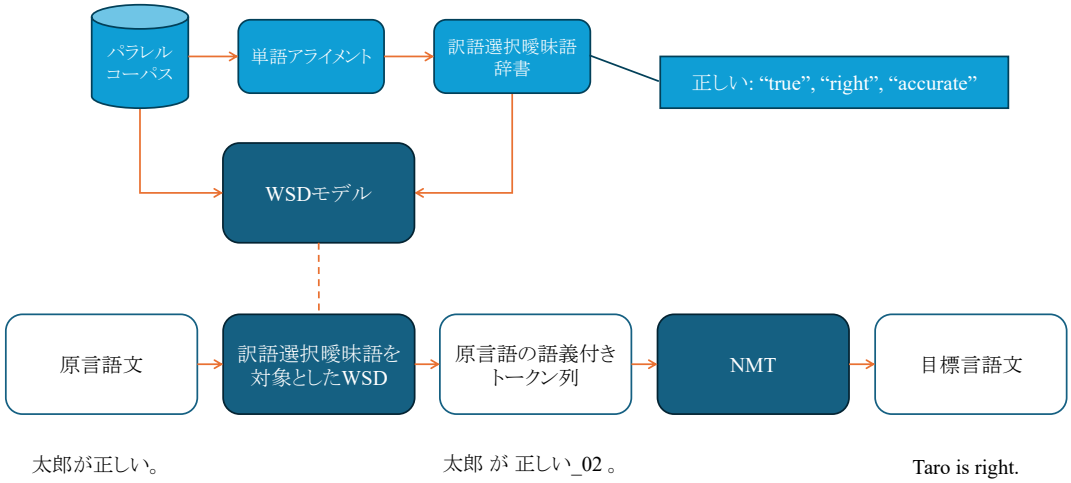


図 3.1: 提案手法の概要

3.2 訳語選択曖昧語辞書の構築

本節では訳語選択曖昧語辞書の構築について述べる。この辞書は訳語選択曖昧語を収録したデータベースであり、訳語選択曖昧語とそれに対する目標言語の訳語のリストの情報を持つ。訳語選択曖昧語辞書は日英パラレルコーパスから自動構築する。日英パラレルコーパスは JParaCrawl(v3.0)[20] を用いる。具体的な手順を以下に示す。

1. **前処理**

   パラレルコーパスに対して単語分割とレンマ化 (活用形の単語を原形に変換する処理) を行う。日本語では MeCab[21](形態素解析辞書として UniDic[22] を使用)、英語では NLTK[23] を用いる。

2. **単語アライメント**

   統計的単語アライメントツール GIZA++[18] を用いて、前処理を行ったパラレルコーパスから単語間の対応関係を推定する。GIZA++[18] は、コーパス全体における単語の共起頻度に基づき、単語ペアの出現頻度や翻訳の確信度を示す翻訳確率を算出する。コーパス全体の単語の対応関係を集約し、以下のフォーマットに直す。

$$
\begin{aligned}
W_1^J\ o_1^J\colon&\quad W_{1,1}^E\ p_{1,1}^E,\ W_{1,2}^E\ p_{1,2}^E,\ \cdots\\
W_2^J\ o_1^J\colon&\quad W_{2,1}^E\ p_{2,1}^E,\ W_{2,2}^E\ p_{2,2}^E,\ \cdots\\
&\ \ \vdots
\end{aligned}
$$

ここで、各記号の意味は以下の通りである。

- W_i^J は i 番目の日本語単語
- o_i^J は W_i^J の出現頻度
- $W_{i,j}^E$ は日本語単語 W_i^J に対応する j 番目の英単語
- $p_{i,j}^E$ は W_i^J が $W_{i,j}^E$ に翻訳される確率

3. **訳語選択曖昧語の選別**

2 で算出した単語の対応関係から訳語選択曖昧語とその訳語のリストを選別し、訳語選択曖昧語辞書を構築する。選別条件は以下の通りである。

訳語選択曖昧語の選別条件

- 出現頻度 $\gamma \geq 20$
- 翻訳候補が 2 個以上かつ 10 個以下

訳語の選別条件

- 翻訳確率 $\alpha \geq 0.1$
- 最大翻訳確率との差 $\beta < 0.2$

例として、訳語選択曖昧語辞書を構築する過程を図 3.2 と 3.3 に示す。図 3.2 は前処理の例を表しており、左は原文、右は単語分割とレンマ化の処理を行った文である。図 3.3 は訳語選択曖昧語の選別を表している。赤線は翻訳確率が α 未満 ($0.00232672 > 0.1$) であるため削除された訳語を表しており、青線は最大翻訳確率との差が β 以上 ($0.317704 - 0.107704 = 0.21 \geq 0.2$) であるため削除された語義を表している。

原文	単語分割+レンマ化
The infrastructure that the world's developing countries need to build and the infrastructure that developed countries need to upgrade will require astronomical amounts of money.	the infrastructure that the world ' s develop country need to build and the infrastructure that developed country need to upgrade will require astronomical amount of money .
世界の開発途上国が構築すべきインフラと、先進国がアップグレードすべきインフラを考えると、必要な資金は天文学的な数字になります。	世 の 開発 途上 国 が 構築 する べし インフラ と 、 先進 国 が アップ グレード する べし インフラ を 考える と 、 必要 だ 資金 は 天文学 的 だ 数字 に なる ます 。

図 3.2: 訳語選択曖昧語辞書構築における前処理の例

原言語の単語	目標言語の単語	確率
利益	profit	0.317704
利益	income	0.235048
利益	interest	0.207055
利益	benefit	0.129335
利益	return	0.107704
利益	revenue	0.00232672

図 3.3: 訳語選択曖昧語辞書構築における訳語の選別の例

最終的に得られた訳語選択曖昧語辞書の統計を表 3.1 に示す。また、訳語選択曖昧語とその訳語の例を図 3.4 に示す。

表 3.1: 訳語選択曖昧語辞書の統計情報

訳語選択曖昧語数	10,009
平均出現頻度	949.86
平均語義数	3.44

予想 2989: forecast 0.283593, expect 0.373676
個性 642: unique 0.191506, individuality 0.195296, personality 0.203853
利益 5769: benefit 0.129335, interest 0.207055, income 0.235048, profit 0.317704

図 3.4: 訳語選択曖昧語辞書 (抜粋)

3.3 WSD モデル

提案手法では、訳語選択曖昧語に対して WSD を行い、語義毎にトークンを分ける処理を行う。本節では WSD モデルの詳細について述べる。

3.3.1 既存の WSD モデルの使用

訳語選択曖昧語の語義を決定するために、WSD ツール KyWSD[24] を用いる。KyWSD は岩波国語辞典を語義の定義として、各単語に語義 ID を付与する全単語語義曖昧性解消システムである。形態素解析器 KyTea をベースとしており、周辺単語を特徴量として用いる点推定により、形態素解析と同時に各単語の語義を推定する。例として、「多くの人が店で買う。」という文に対して KyWSD で WSD を行い、語義毎にトークンを細分化した結果を以下に示す。

多くの人が店で買う。
↓
多く_5174-0-0-0 の_0 人_43399-0-0-1 が_0 店_31804-0-0-1 で_0 買_7225-0-0-1 う_0 。_0

KyWSD では文の単語分割も行う。スペースは単語の区切りを表す。また、5174-0-0-0 といった記号はそれぞれの単語に対して推定された語義の ID を表す。語義 ID を「_」で元の単語と連結することでトークンを語義毎に細分化している。なお、「の」「が」「で」といった助詞や「う」といった語尾などの機能語については WSD は行わない。これらの単語に付与された「0」という語義 ID は機能語に対して付与するダミーの語義 ID である。

3.3.2 訳語を語義とする WSD モデル

KyWSD は岩波国語辞典を語義の定義としているが、辞書の語義は英語への翻訳と直接関係しているわけではない。すなわち、辞書における異なる語義が英語の同じ単語に翻訳されることもあれば、同じ語義でも別の英単語に翻訳されることもある。したがって、KyWSD を用いた語義毎のトークンの細分化は訳語選択曖昧語の訳し分けの性能向上に貢献しない可能性もある。

そこで、目標言語の訳語を語義とする WSD を行い、トークンを分ける処理を行う。すなわち、訳語選択曖昧語に対し、それが翻訳されるべき英単語を語義として推定する WSD モデルを学習し、それをもとにトークンを細分化する。本項ではこの WSD モデルの学習の詳細について述べる。

訓練データの構築

パラレルコーパスにおける単語アライメント結果から、対象単語に対して正解の語義を付与したデータセットを構築する。まず、訳語選択曖昧語が与えられたとき、日英パラレルコーパスからそれを含む日本語文を検索する。次に、検索された日本語文と対になっている英語文において、訳語選択曖昧語に対する複数の語義 (訳語) のうち一つのみが出現しているとき、訳語選択曖昧語の語義が一意に定まると仮定して、日本語文にその語義を正解としてラベル付けする。この操作

を訳語選択曖昧語毎に繰り返す。すなわち、個々の訳語選択曖昧語毎に正解の語義がラベル付けされた訓練データを構築する。

上記の手続きの際、文中に同一の対象単語が 2 回以上出現する文は訓練データから除外する。その理由は、文中に複数の対象単語が存在する場合、どの出現位置の単語がどの訳語に対応しているかの対応関係を一意に特定することが困難なためである。以下の例で考えると、1 番目の「利益」と 2 番目の「利益」がそれぞれ “profit” と “income” のどちらに対応しているかを自動的に判断することは難しく、誤った語義を付与する可能性がある。

日文: 利益率が上がり、純利益も上がった。
英文: Profit margins increased, and net income also rose.

WSD の対象単語の設定

WSD モデルは予測するクラス (語義) が単語毎に異なるため、単語毎に個別の WSD モデルを学習する必要がある。しかし、本研究における訳語選択曖昧語辞書ではおよそ 10,000 語の訳語選択曖昧語があり、その全てに対して WSD モデルを学習するのは困難である。そのため、WSD の対象とする単語を限定して WSD モデルを学習する。具体的には、まず極端な高頻度語や低頻度語を除外するため、パラレルコーパスにおける出現頻度が 200 未満もしくは 20,000 より大きい訳語選択曖昧語を除外する。次に、前述の手続きで十分な量の訓練データが確保できる単語、具体的には訓練データのサンプル数が 500 以上の訳語選択曖昧語を選別する。これにより、1725 単語を WSD モデル学習の対象とする訳語選択曖昧語として選定した。

WSD モデルの学習

本研究では、東北大で構築・公開されている日本語事前学習済み BERT[25] をベースに WSD モデルを構築する。また、WSD モデルとして以下の 3 つを実装する。各モデルの構造 (アーキテクチャ) を図 3.5 に示す。

- **BERT-pre:** 事前学習済み BERT に FCL 層を追加し、FCL 層のみ学習したモデル。FCL 層は単語毎に用意し、単語毎に異なる語義クラスを出力させるとともに、単語に固有の情報を学習させる。BERT のパラメタは更新せず、FCL 層のパラメタのみ更新するため、学習時間は比較的短い。
- **BERT-ft-shared:** 全単語に共通の事前学習済み BERT に単語毎の FCL 層を追加して学習したモデル。BERT-pre とは異なり、BERT のパラメタも WSD の学習データによってファインチューニングする。そのため学習に要する時間が長くなる。

- **BERT-ft-indi:** 単語毎に事前学習済み BERT + FCL 層を学習したモデル。BERT モデルも単語毎に個別のものを用意し、WSD の学習データによるファインチューニングも行う。単語毎に完全に別々のモデルを用意するため、各単語に特化した情報を精緻に学習することが期待できるが、WSD モデル全体で多くのディスク容量を必要とし、学習時間も長いというデメリットがある。

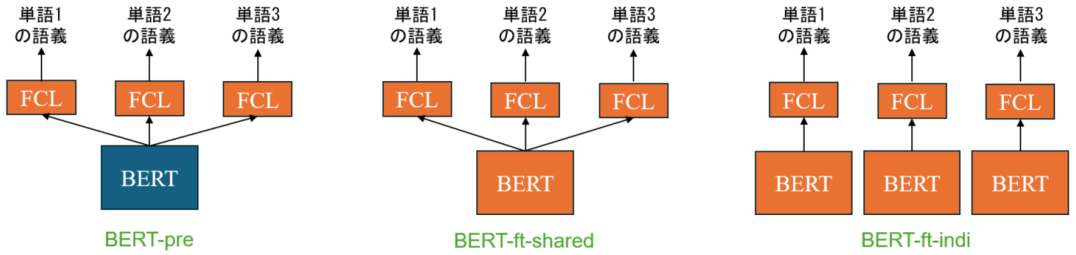


図 3.5: 3 つの WSD モデルのアーキテクチャ

3.3.3 unknown クラスの導入

3.3.2 項で説明した訳語を語義とする WSD では、訳語選択曖昧語に対して訳語選択曖昧語辞書に含まれる語義 ID のいずれかを必ず付与する必要があった。しかし、実際の翻訳タスクにおいて、対象語の正解訳が訳語選択曖昧語辞書に含まれる語義のいずれにも該当しないケース（辞書外の語義）が存在する。このようなケースにおいて、訳語選択曖昧語辞書内の既存の語義 ID を強制的に付与することは、翻訳モデルに誤った語義の情報を与えることとなり、翻訳モデル学習におけるノイズとなりうる。

この問題を解決するため、辞書に含まれない語義を表す unknown クラスを導入する。ある訳語選択曖昧語 w が持つ訳語選択曖昧語辞書で定義された語義の集合を $S_w = \{s_1, s_2, \ldots, s_n\}$ とする。ここまでの WSD タスクが w を S_w のいずれかに分類する問題であったのに対し、unknown クラス s_{unk} を追加した語義の集合 $S'_w = S_w \cup \{s_{unk}\}$ を定義し、w を S'_w のいずれかのクラスに分類するタスクとして定式化する。これにより、モデルが辞書内の語義か、あるいは辞書外の語義かを区別して予測する。

Unknown クラスを推定する手法として、2 つのアプローチを提案する。1 つは、(1)unknown タグを付与した訓練データを作成し、WSD モデルに unknown という独立したクラスとして直接推定させる方法、もう 1 つは (2)unknown を含まない通常のデータセットでモデルを学習させ、推論時にモデルの確信度が一定の閾値を下回った場合を unknown と判定する方法である。本研究では閾値を 0.65 と設定する。(1) のとき、訳語選択曖昧語辞書外の語義を持つ用例を JParaCrawl[20] から最

大 1000 文ランダム抽出し、先に述べた WSD の訓練データ追加したデータセットを作成し、WSD モデルを訓練する。

(1)、(2) の例を以下に示す。

- **学習時**

  以下は単語「利益」の WSD モデルを学習するための訓練データを表わす。末尾の [ ] で囲まれた数字は語義 ID を表しており、パラレルコーパスにおける参照訳に基づいて正解の語義 ID が与えられている。また、括弧内が数字の時は訳語選択曖昧語辞書内の語義 ID であることを、unknown の時は訳語選択曖昧語辞書外の語義であることを表す。(1) では語義 ID が付与された例 a に加え、unknown が付与された例 b を含むデータセットを用いて WSD モデルを学習する。一方 (2) では語義 ID が付与された例 a のみで構成されるデータセットを用いて WSD モデルを学習する。

  WSD 対象単語: 利益

  (1)

  a. これは主に利益剰余金の減少によるものです。 [3]

  b. 営業利益は自動ドア関連の競争激化に伴う採算悪化等により、減益となりました。 [unknown]

  (2)

  a. これは主に利益剰余金の減少によるものです。 [3]

- **推論時**

  以下は学習した WSD モデルを用いて単語「利益」の語義を推定する過程を表す。(1) の場合は、unknown をクラスとして考え、語義の推定確率が最も高いものを出力する。(2) の場合は、語義の推定確率が最も高いもので 0.65 未満である場合、unknown と出力する。

  WSD 対象単語: 利益

  (1)

  これらは利益や配当金の送金するに関わる制限だけでなく、為替調整や輸出入許可も含みます。 [unknown]

  語義の推定確率 {profit: 0.30, income: 0.10, interest: 0.15, benefit: 0.10, unknown: 0.35}

  (2)

  またそれらの組織でも同じ点に重点を置き、効果的に利益を伸ばしている。 [unknown]

  語義の推定確率 {profit: 0.40, income: 0.20, interest: 0.25, benefit: 0.15}

3.3.4 uncertain クラスの導入

unknown クラスでは訳語選択曖昧語辞書外の語義であることを推測させたが、実際の翻訳タスクにおいては、訳語選択曖昧語辞書外の語義である場合だけでなく、文脈情報が少なく語義の特定が困難な場合や、複数の語義が該当する場合など、対象単語の語義を 1 つに決めること事態が困難な事例もある。このような事例に対し、無理に一つの語義 ID を割り当てることは、翻訳モデルに対してノイズを与えることになると考えられる。

この問題を解決するため、モデルの確信度が低いことを表す uncertain クラスを導入する。訳語選択曖昧語 w の語義集合 S_w に対し、uncertain クラス s_{uncert} を追加した $S''_w = S_w \cup \{s_{uncert}\}$ を定義する。ここで重要な点は、uncertain クラスがカバーする範囲である。モデルが自信を持てないケースとは、文脈が難解である場合に加え、正解が辞書に含まれていない場合 (unknown) も含まれる。なぜなら、正解の語義を知らないモデルは、既存のどの語義に対しても高い確信度を持つことができないからである。したがって、uncertain クラスは語義の識別が困難な場合と unknown クラスを内包した広範なクラスであると言える。

本研究では、uncertain クラスの推測にはモンテカルロドロップアウト (Monte Carlo Dropout; MCD) を用いる。MCD は、推論時にも Dropout を有効にすることでモデルの内部状態に意図的な揺さぶりを与え、その出力変動を観測する手法である。もし入力に対してモデルが十分な学習をしており、明確な正解が存在するならば、多少の揺さぶりを与えても予測結果は変化しないが、判断が難しい単語や未知の語義に対しては、ドロップアウトによって予測結果が大きくばらつくことが予想される。MCD の具体的な手順は以下の通りである。

1. 同一の入力文に対して、Dropout を適用した状態で N 回の推論を行う。
2. N 回の出力結果を集計し、最も多く支持された語義の出現回数 C_{max} を算出する。
3. C_{max} が閾値 T 未満であれば、モデルの確信度が低いとみなし、最終出力を s_{uncert} とする。
4. $C_{max} \geq T$ であれば、その語義を信頼できる予測として採用する。

本研究では $N = 10$ とし、同一語義が 8 回以上 ($T = 8$) 予測されなかった場合を uncertain と判定する。

3.4 翻訳モデルの学習

翻訳モデルは OpenNMT[26] を用いて Transformer モデル [19] をベースとした NMT モデルを学習する。OpenNMT[26] とは、前処理、学習、推論という NMT の一連のパイプラインを包括的に提供するオープンソースのツールキットである。

本研究では、既存の事前学習済みモデルを用いたファインチューニングではなく、モデルをスクラッチから学習する。スクラッチ学習を行う主な理由は、WSDモデルによって付与される独自の語義の ID を適切に処理するためである。事前学習済みの翻訳モデルは、一般的なコーパスで構築された固定の語彙辞書を持っている。しかし、本研究で導入する語義 ID(unknown クラスや uncertain クラスを含む) によって細分化されたトークンは、一般的な翻訳モデルの語彙には含まれない特殊なトークンである。スクラッチから学習を行うことで、これらの独自トークンを語彙に含め、翻訳モデルがそれらを認識させることができる。これにより、翻訳モデルは語義 ID と訳語の対応関係を他の一般的な単語と同様に埋め込み表現として学習し、文脈に応じた訳語選択曖昧語の訳語を生成することができる。

第4章　実験・評価

本章では提案手法の評価実験について述べる。評価実験に用いたデータセット、実験設定、実験結果と考察、ケーススタディとエラー分析について述べる。

4.1　実験設定

4.1.1　データセット

翻訳モデルの学習と評価には日英パラレルコーパス JParaCrawl(v3.0)[20] を用いる。JParaCrawl からランダム抽出した 100 万文対を訓練データとして使う。翻訳モデルの学習が正常にできているか確認のための検証データ 5000 文、評価用のテストデータ 4000 文も同様に JParaCrawl からランダムに抽出する。

WSD モデルの学習と評価には、対象となる 1725 語の訳語選択曖昧語を含む文を抽出したデータセットを用いる。このデータセットは 3.3.2 項で述べた手法で構築する。また、以下の 2 種類のデータセットを用意する。

- **Standard:** 訳語選択曖昧語辞書内の語義を持つ用例のみで構成されるデータセット。
- **Unknown:** Standard に加え、訳語選択曖昧語辞書外の語義 (unknown クラスを持つ用例) を持つ用例を最大 1000 文追加したデータセット。

表 4.1 に WSD 訓練データの統計情報を示す。表中の「平均語義数」は対象単語が持つ語義の数の平均であり、「平均サンプル数」は 1 単語あたりの学習に使用された文の平均数である。また、評価用のテストデータについても、訓練データと同じ方法で構築し、各単語で訓練データとは重複しない 200 文をランダムに抽出する。

表 4.1: WSD 訓練データの統計情報

データセット	対象単語数	平均語義数	平均サンプル数
Standard	1,725	2.58	1,799.62
Unknown	1,725	2.58	2,792.91

4.1.2 実験設定

1. **比較する手法**

WSD モデルでは、以下の 3 手法を比較する。

- **BERT-pre:** 事前学習済み BERT に FCL 層を追加し、FCL 層のみ学習したモデル。
- **BERT-ft-shared:** 全単語に共通の事前学習済み BERT に単語毎の FCL 層を追加して学習したモデル。
- **BERT-ft-indi:** 単語毎に事前学習済み BERT + FCL 層を学習したモデル。

翻訳モデルでは、以下の 10 手法を比較する。

- **Vanilla (Baseline 1):** 分かち書きのみ行った生テキストコーパスを用いて訓練した NMT モデル。WSD は行わない。ベースラインのひとつ。
- **JDIC-all (Baseline 2):** 全単語に対して WSD を行い、語義毎のトークンを分けたコーパスを用いて訓練した NMT モデル。WSD は KyWSD で行う。先行研究 [2] の再現。ベースラインのひとつ。
- **JDIC-amb:** 訳語選択曖昧語に対してのみ WSD を行い、語義毎にトークンを分けたコーパスを用いて訓練した NMT モデル。WSD は KyWSD を用いる。
- **TR-pre:** 訳語選択曖昧語に対してのみ WSD を行い、語義毎にトークンを分けたコーパスを用いて訓練した NMT モデル。WSD のモデルは、事前学習済み BERT に FCL 層を追加し、FCL 層のみ学習したモデルを用いる。
- **TR-pre-unk-train:** TR-pre を基本設定とし、WSD のクラスとして unknown を追加する。unknown クラスがタグ付けされた訓練データ (表 4.1 における Unknown データセット) から WSD モデルを学習する。
- **TR-pre-unk:** TR-pre を基本設定とし、WSD のクラスとして unknown を追加する。WSD の判定の信頼度が低いときに unknown と判定する。
- **TR-ft-shared:** 訳語選択曖昧語に対してのみ WSD を行い、語義毎にトークンを分けたコーパスを用いて訓練した NMT モデル。WSD のモデルは、全単語に共通の事前学習済み BERT に単語毎の FCL 層を追加して学習したモデルを用いる。
- **TR-ft-shared-unk:** TR-ft-shared を基本設定とし、WSD のクラスとして unknown を追加する。WSD の判定の信頼度が低いときに unknown と判定する。

- **TR-ft-shared-uncer:** TR-ft-shared を基本設定とし、WSD のクラスとして uncertain クラスを追加する。WSD の判定の信頼度が低いときに uncertain と判定する。
- **TR-gold:** 訳語選択曖昧語に対してのみ WSD を行い、語義毎にトークンを分けたコーパスを用いて訓練した NMT モデル。WSD は 100%正しく推論できる理想的なモデルを用いる。テスト文の日本語文に対し、それに対応する英語文の中に含まれる訳語の候補をチェックし、正解の語義 (訳語) を決める。

上記 10 個の手法の特徴を表 4.2 にまとめる。

表 4.2: 比較する翻訳モデルのまとめ

	語義	WSD モデル	PRO	UNK	UNC
Vanilla	–	–			
JDIC-all	辞書	KyWSD			
JDIC-amb	辞書	KyWSD	✓		
TR-pre	訳語	BERT-pre	✓		
TR-pre-unk-train	訳語	BERT-pre	✓	✓	
TR-pre-unk	訳語	BERT-pre	✓	✓	
TR-ft-shared	訳語	BERT-ft-shared	✓		
TR-ft-shared-unk	訳語	BERT-ft-shared	✓	✓	
TR-ft-shared-uncer	訳語	BERT-ft-shared	✓		✓
TR-gold	訳語	–	✓		

「語義」は WSD モデルの語義の定義が辞書 (KyWSD が採用している岩波国語辞典) もしくは訳語であるかを表す。「PRO」は提案手法、すなわち訳語選択曖昧語のみ WSD を行い語義毎にトークンを細分化するモデルを表す。「UNK」、「UNC」はそれぞれ unknown クラス、uncertain クラスを導入していることを表す。

2. **評価基準**

WSD モデルの評価指標として正解率を用いる。

翻訳モデルの評価には以下の 2 つの指標を用いる。

- **BLEU[27]:** 翻訳の品質を自動評価する代表的な指標である。参照文 (正解の翻訳文) とモデルが翻訳した文の単語 n-gram の重複を測ることでスコアを算出する。本実験では SacreBLEU[1] を用いて BLEU スコアを算出する。

[1] `https://github.com/mjpost/sacrebleu`

- **訳語選択正解率:** 訳語選択曖昧語が正しい訳語に翻訳されているかを測るため、独自の評価指標を用いる。本評価では、訳語選択曖昧語辞書に定義された語義がリファレンス (参照訳) に存在するとき、その語義を正解とみなし、モデルによる出力の中にその語義が含まれていれば、その訳語選択曖昧語が正しく翻訳されたとみなす。訳語選択正解率は、基本的には正しく翻訳された訳語選択曖昧語の割合であるが、正解の判定基準が異なる 2 つの指標を用いる。
  - **正解率:** モデルの出力文中に、辞書に定義されている語義がただ 1 語のみ含まれており、かつその語義がリファレンスの語義と一致した場合を正解とする。翻訳モデルが文脈から正しい語義を一意に特定できているかを測る指標である。
  - **準正解率:** モデルの出力文中に、リファレンスの語義が含まれている場合を正解とする。この際、出力文中に辞書にある他の語義が同時に含まれていても、正解の語義が含まれていれば準正解とみなす。出力に不正解の語義が含まれていても、正しい語義を取りこぼさずに翻訳できているかを測る指標である。

  正解、準正解、不正解の例を以下に挙げる。

  評価対象文: 「彼の指導は甘い。」

  辞書にある「甘い」の訳語候補: [“sweet”, “naive”, “lenient”]

  正解 (リファレンス) の英単語: “lenient”

  - 正解の例
    モデル出力: “His advice is ***lenient***.”
  - 準正解の例
    モデル出力: “His advice is ***lenient*** and ***naive***.”
  - 不正解の例
    モデル出力: “His advice is ***sweet***.”

  また、訳語選択正解率における評価対象語を以下の 2 種類に分類し、評価を行う。

  - **Target Words:** WSD の対象とした (WSD モデルを学習した) 訳語選択曖昧語のみに対する訳語選択正解率
  - **All Words:** WSD モデルの学習の有無にかかわらず、全ての訳語選択曖昧語に対する訳語選択正解率

3. **学習パラメータ**

   翻訳モデルは基本的には OpenNMT[26] の例に従って学習を進めた。予備実験により、WSD モデルにおいて、BERT から得られる埋め込みから語義のクラスを予測するモジュールとして 1 層の FCL と 2 層の FCL を比較したが、

2 層の FCL を用いた方が WSD の正解率が高かったため、本実験では 2 層の FCL を採用した。バッチサイズと勾配累積のパラメータは計算資源の制約と学習の安定性を考慮して設定した。その他のパラメータについては、一般的なモデル学習で広く用いられている値を採用した。学習パラメタの詳細を以下に示す。

- **WSD モデル**
  エポック: 8
  バッチサイズ: 8 (sentences)
  勾配累積: 4
  学習率: 2e-5
  重み減衰: 0.01
  FCL 層: 2 (768 次元 → ReLU 関数 → 128 次元)
- **翻訳モデル**
  訓練ステップ数: 50000 ステップ (40 エポック程度)
  ウォームアップステップ数: 8000 ステップ
  バッチサイズ: 4096 (tokens)
  勾配累積: 8
  エンコーダー・デコーダー層: 6 層
  隠れ層サイズ: 512
  ヘッド数: 8
  学習率: 2.0
  ドロップアウト率: 0.1
  最適化手法: Adam
  学習率のスケジューリング: Noam Decay
  ラベル平滑化: 0.1

4.2 実験結果と考察

4.2.1 WSD モデルの評価

目標言語の単語を語義とし、本研究で独自に学習した WSD モデルの評価を行った。結果を表 4.3 に示す。実験の結果、いずれのデータセットにおいても、事前学習済み BERT をそのまま用いるモデル (BERT-pre) より、ファインチューニングを行ったモデル (BERT-ft-shared、BERT-ft-indi) の方が高い正解率を示した。Unknown データセットにおける結果を見ると、BERT-ft-indi が BERT-ft-shared

を 0.32 ポイント上回っている。しかし、BERT-ft-indi の構築は WSD の対象とした 1725 単語の BERT モデルを個別に学習する必要があり、膨大な計算時間を要する。Unknown データセットでの比較において、BERT-ft-shared との精度差はそれほど大きくなく、計算コストに見合う大幅な性能向上は見込めないと判断したため、Standard データセットでの学習は実施しなかった。以上の結果を踏まえ、翻訳モデルの評価実験では BERT-pre と BERT-ft-shared を用いる。

表 4.3: WSD モデルの評価結果

モデル	Standard	Unknown
BERT-pre	0.8278	0.6755
BERT-ft-shared	**0.8488**	0.6951
BERT-ft-indi	—	**0.6983**

4.2.2 翻訳モデルの評価

翻訳実験の結果を表 4.4 に示す。また、全訳語選択曖昧語もしくは学習した訳語選択曖昧語のみを対象とした訳語選択正解率・準正解率をグラフに示したものを図 4.1 と 4.2 に示す。分かりやすさのため、全訳語選択曖昧語もしくは学習した訳語選択曖昧語のみを対象とした訳語選択正解率のみを示したグラフを付録 A の図 A.1 と A.2 にそれぞれ示す。表 4.4 を見ると、JDIC-amb やベースラインの JDIC-all は Vanilla と比べて、訳語選択正解率はほとんど変わらないものの、BLEU はそれぞれ 0.7 ポイント、0.9 ポイント下回っている。一方で、訳語選択曖昧語の訳語を語義の定義とした WSD モデルを用いたモデル (TR-*) は一貫して良い結果が出ている。既存の WSD ツール KyWSD を用いた JDIC-all や JDIC-amb の翻訳性能が低いのは、KyWSD の WSD の正解率が低い可能性が考えられる。KyWSD が比較的文法的に正しい文で訓練されているのに対し、本研究で実験に使用した JParaCrawl[20] では文法的な誤りや断片的な文、スラングなどが含まれており、KyWSD が正確に語義を推定できなったと推察される。また、KyWSD は辞書を語義の定義としているが、辞書の語義が翻訳における訳語選択の曖昧性と直接的に関連付けられていないことも原因と思われる。

本研究では、WSD を行う際に、語義ラベル以外に unknown クラスや uncertain クラスを導入し、WSD モデルが事前に定義された語義の識別ができない状況に対応しようとしたが、これらのモデルは追加の語義のクラスを使わない TR-ft-shared よりも性能が低かった。また、unknown クラスを事前に訓練コーパスに付与した状態で訓練させた TR-pre-unk-train は訳語選択正解率では最も精度が低かった。これは、unknown クラスと uncertain クラスの導入が有効ではないことを示している。

表 4.4: 提案手法ならびにベースラインの翻訳モデルの評価

Model	**BLEU**	**Target Words**		**All Words**	
		Strict	**Semi**	**Strict**	**Semi**
Vanilla	27.9	0.523	0.676	0.522	0.674
JDIC-all	27.0	0.527	0.671	0.522	0.674
JDIC-amb	27.2	0.523	0.670	0.520	0.669
TR-pre	27.9	0.530	0.680	0.524	0.674
TR-pre-unk-train	**28.1**	0.520	0.668	0.520	0.671
TR-pre-unk	**28.1**	0.527	0.678	0.522	0.674
TR-ft-shared	**28.1**	**0.534**	**0.686**	**0.531**	**0.683**
TR-ft-shared-unk	27.9	0.526	0.672	0.518	0.667
TR-ft-shared-uncer	27.9	0.531	0.682	0.526	0.679
TR-gold	28.9	0.689	0.860	0.630	0.793

Strict は訳語選択正解率を、Semi は訳語選択準正解率を表す。Target Words と All Words はそれぞれ、WSD モデルが学習対象とした訳語選択曖昧語と、学習の有無にかかわらず全ての訳語選択曖昧語を評価対象語とした場合を表す。

TR-ft-shared は最も良い性能を示しているが、Vanilla と比べて大きな性能向上は見られなかった。しかし、訳語選択曖昧語に対し、完全に正解のタグを付けられた TR-gold は Vanilla と比べ、BLEU が 1 ポイント、WSD モデルを学習した訳語選択曖昧語のみを対象とした訳語選択正解率が 16.6 ポイント (約 31.7%)、準正解率が 18.4 ポイント (約 27.2%) 向上し、大幅な改善が確認された。このことは、訳語選択曖昧語に限って WSD を行い、語義毎にトークンを細分化した上で翻訳モデルを学習するという提案手法のアプローチが妥当であることを示唆している。

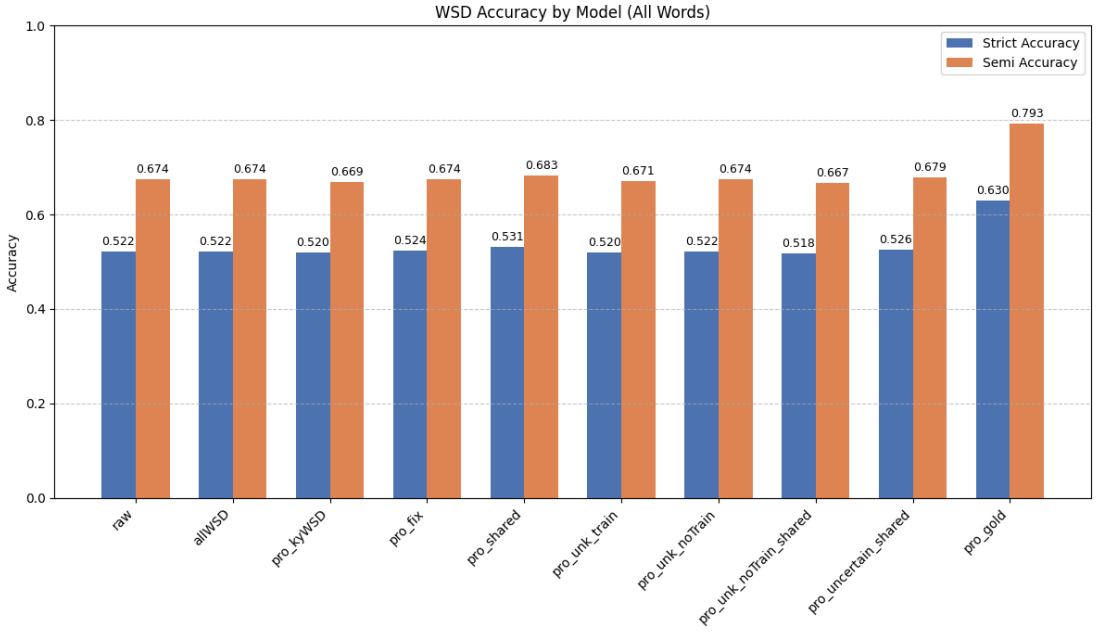


図 4.1: 全訳語選択曖昧語を対象とした訳語選択正解率・準正解率

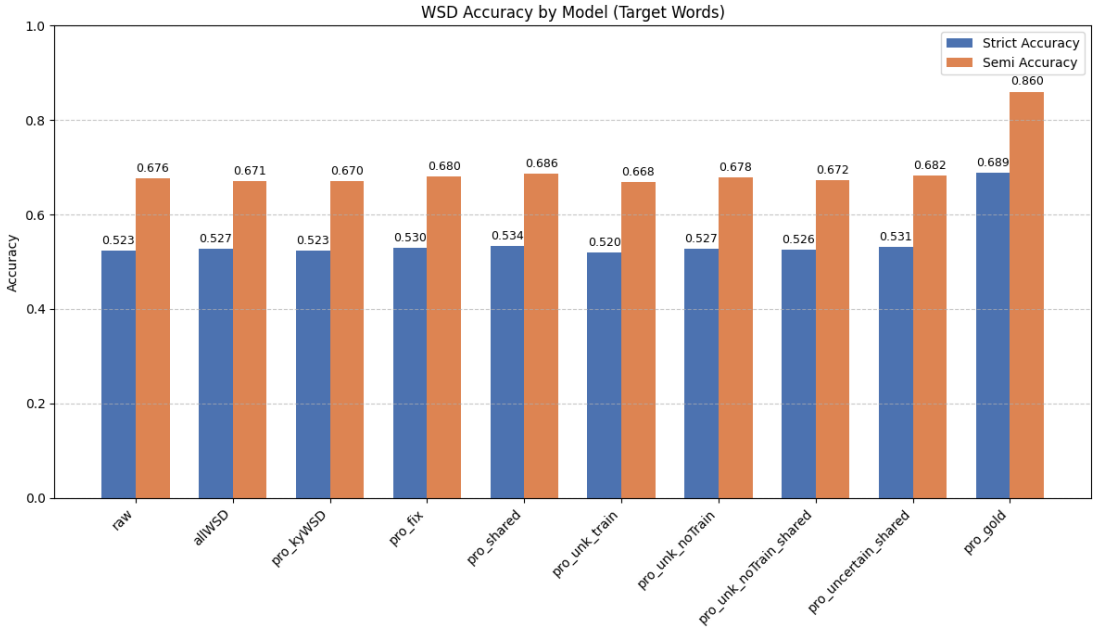


図 4.2: 学習した訳語選択曖昧語のみを対象とした訳語選択正解率・準正解率

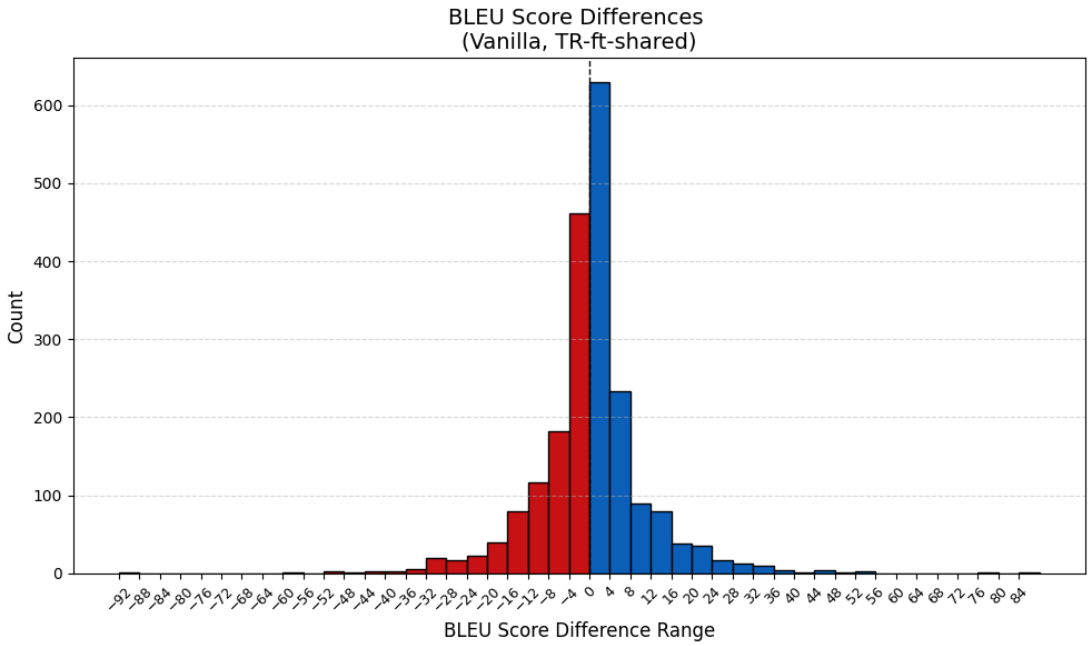
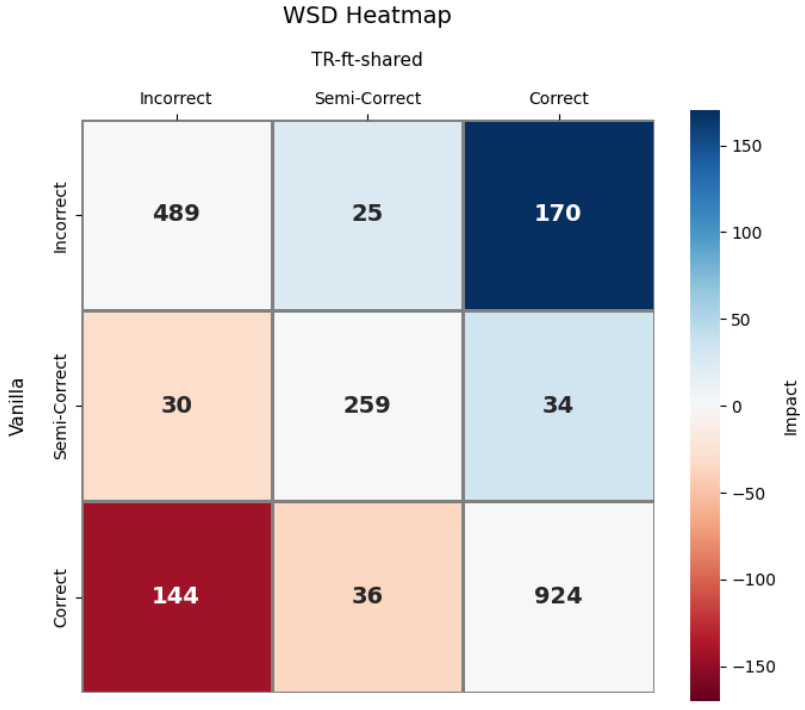


図 4.3: 個々の文における TR-ft-shared と Vanilla の BLEU スコアの差の分布



Correct は正解、Semi-Correct は準正解、Fail は不正解を表す。青色は改善、赤色は悪化、白は変化なしを表す。色の濃さは件数の多さを表す。

図 4.4: Vanilla と TR-ft-shared の訳語選択曖昧語に対する訳語の正解判定結果の対応表

次に、Vanilla と提案手法で最も性能が高かった TR-ft-shared を詳細に比較する。図 4.3 は TR-ft-shared と Vanilla の文ごとの BLEU スコアの差の分布を示したものである。差は TR-ft-shared の BLEU スコアから Vanilla の BLEU スコアを引いて算出している。横軸は BLEU スコアの差を 4 ポイントごとに区切ったものであり、縦軸は各範囲に該当する文の数である。青色の領域は Vanilla と比べて TR-ft-shared の方が BLEU スコアが高く、赤色の領域は低いことを表している。この図から、TR-ft-shared が Vanilla と比べて、BLEU が 0 から 8 ポイント改善した文が同じく 0〜8 ポイント悪化した文より多くあることが示されている。一方で、それ以上の BLEU の差については、基本的に悪化した文の方が多いことが確認できる。しかし、極端な BLEU の改善、悪化が少ないことから、提案手法は副作用が少なく、局所的に訳語選択の誤りを改善する安定した手法であると言える。

図 4.4 は各文の訳語選択曖昧語の翻訳結果を「正解」(正しい訳語に一意に翻訳された場合)、「準正解」(正しい訳語に翻訳されたが他の訳語候補も訳出された場合)、「不正解」(正しい訳語に翻訳されてない場合) の 3 つに分類したときの Vanilla と TR-ft-shared の対応表である。また、より詳細な対応表を付録 A の図 A.3 に示す。TR-ft-shared は正解したが Vanilla は不正解だった文が 170 件、TR-ft-shared は不正解だったが Vanilla は正解した文が 144 件あり、結果として TR-ft-shared の方が正解の文が 26 件多いことがわかる。一方で、Vanilla では準正解、TR-ft-shared では正解の文の数が、TR-ft-shared では準正解、Vanilla では正解の文の数を 2 件下回っている。また、Vanilla では不正解、TR-ft-shared では準正解の文の数が、TR-ft-shared では不正解、Vanilla では準正解の文の数を 5 件下回っている。提案手法は Vanilla と比べて、準正解の文は少ないが正解の文が多くあることが示されており、一意に語義を推定するのに適した手法だと言える。

4.3 翻訳例

本節では、実際の翻訳例を対象とした定性的な分析を行うことで、提案手法が具体的にどのように訳語選択曖昧語の訳し分けを改善したか、またどのような課題があるのかをより深く考察する。具体的には改善が見られた事例を分析するケーススタディと、誤りの原因を分析するエラー分析を行う。

4.3.1 ケーススタディ

ベースライン (Vanilla) では正しい訳語に翻訳できなかったが、提案手法 (TRft-shared) では翻訳できた例を 3 つ挙げる。以下の例では、Ref は参照訳を表す。

1. 対象単語: 取引

   正解の訳語: trading

日文: 過去 3 年間 で 、 昨年 達成 さ れ た 最高 の **取引** 高 は 、 推定 400 万 株 です 。

Ref: Over the past three years , the highest **trading** volume achieved in last year , estimated four million shares .

Vanilla: The best **deal** reached last year in the past three years is an estimated 4 million shares . [BLEU:20.4]

TR-ft-shared: In the past three years , the highest **trading** volume achieved last year is an estimated 4 million shares . [BLEU:56.8]

考察:

Vanilla では “deal”、TR-ft-shared では “trading” と翻訳されている。どちらも「取引」の訳語として一般的だが、“deal” は個人間の契約やビジネス上の商談で使われる単語であり、Vanilla の翻訳文だと「一回の取引で 400 万株を動かした」というニュアンスになる。ここでは、年間の総量を表す「取引高 (trading volume)」という専門用語に訳すのが正しいため、“trading” が適切である。このように、提案手法は金融ドメイン特有のコロケーションを考慮し、専門用語を正しく推定できている。

2. 対象単語: すぐ

   正解の訳語: soon

   日文: 秋田 駅 を 出発 し 都市 部 を 抜ける と 、 家 や 建物 の 数 より も 田 ん ぼ の 数 が 多い 田園 風景 が **すぐ** に 広がる 。

   Ref: The train leaves the city from Akita Station and the landscape **soon** opens up into the countryside , where rice paddies outnumber houses and buildings .

   Vanilla: When you leave Akita station and exit the city , you will see a rural landscape with more rice paddies than the number of houses and buildings . [BLEU:12.8]

   TR-ft-shared: When you leave Akita Station and exit the city , you will **soon** see a countryside with more rice fields than the number of houses and buildings . [BLEU:15.0]

   考察: この事例では、“soon” があるかないかだけが違いとしてあり、その他の単語はすべて同じである。これは、語義毎にトークンを区別する提案手法が曖昧な単語を適切に翻訳するだけでなく、訳抜けを防ぐことにも効果的であることを示している。すなわち、語義毎に区別されたトークンを翻訳モデルが重要度の高い単語として認識し、結果として訳語が出力されやすくなっていると考えられる。

3. 対象単語: 復興

   正解の訳語: recovery

   日文: 僕 たち 「 NPO 法人 チーム ふく しま 」 は 、 その 2 カ月 後 に 「 福島 ひまわり 里 親 プロジェクト 」 を 立ち上げ 、 **復興** の ため 、 そして 風 化 対策 の ため に 、 活動 を 続け て き ま し た 。

   Ref: Two months after the earthquake , we launched the Fukushima Sunflower Project and have worked to bring **recovery** to Fukushima .

   Vanilla: We , the NPO team Fukushima , launched the Fukushima Sunflower Parent Project two months later , and continued activities for **reconstruction** and [unk] measures . [BLEU:10.8]

   TR-ft-shared: Two months later , we launched the Fukushima Sunflower Parent Project and have continued our activities for **recovery** and mitigation . [BLEU:32.3]

   考察:

   Vanilla では “reconstruction”、TR-ft-shared では “recovery” と翻訳されている。どちらも「復興」の訳語として一般的だが、“reconstruction” は建物やインフラなどの再建というような意味合いで使うことが多く、“recovery” は経済回復や健康の回復といった社会的・精神的な回復という意味合いで使うことが多い。この一文では読み取り辛いが「NPO 法人」や「里親」、「風化対策」といった単語から精神的な意味としての「復興」が強いと判断できるため、“recovery” が適切である。このようにニュアンスを読み取るのが難しい文でも提案手法は適切な訳語を出力できている。

4.3.2 エラー分析

ベースライン (Vanilla) では正しい訳語に翻訳できたが、提案手法 (TR-ft-shared) では翻訳できなかった例を 3 つ挙げる。

1. 対象単語: 中毒

   正解の訳語: intoxication

   日文: 嘔 吐 衝動 は 胎児 の 発達 に よる **中毒** の 症状 です 。

   Ref: Vomit urge is a symptom of **intoxication** due to the development of the fetus . [BLEU:74.0]

   Vanilla: [unk] impulse is a symptom of **intoxication** due to the development of the fetus .

TR-ft-shared: The vomit urge is a symptom of **poisoning** due to the development of the fetus . [BLEU:67.4]

考察:

Vanilla では “intoxication”、TR-ft-shared では “poisoning” と翻訳されている。どちらも「中毒」の訳語として一般的だが、“poisoning” は食中毒や毒物摂取など、外部からの有害物質によって引き起こされる「中毒」のことを指し、“intoxication” は生体内での代謝変化や特定の疾患によって引き起こされる「中毒」という意味合いで使われる。ここでは、「胎児の発達」によるものと書かれているため、“intoxication” が適切である。訓練データでは “intoxication” を含む文は 76 文しかなく、“poisoning” の半分しかないため、高頻度語である “poisoning” を選択したと考えられる。このように正しい訳語の選択に高度なドメイン知識が必要な単語においては、改善の余地があることを示している。

2. 対象単語: 変形

   正解の訳語: deformation

   日文: これ は 、 窒 化 プロセス 中 に わずか な **変形** しか 発生 し ない た め です 。

   Ref: This is because only a small amount of **deformation** occurs during the nitriding process .

   Vanilla: This is because only a slight **deformation** occurs during the ¡unk¿ process . [BLEU:44.8]

   TR-ft-shared: This is because only a slight **variant** occurs during the ¡unk¿ process . [BLEU:36.2]

   考察:

   Vanilla では “deformation”、TR-ft-shared では “variant” と翻訳されている。どちらも「変形」の訳語として一般的だが、“variant” はソフトウェアのバージョンや生物学的な個体差を指す際に「変種」や「変異体」といった意味合いで使われることが多く、“deformation” は物理的な力や熱によって物体の形状が変わる「歪み」や「奇形」といった意味合いで使われる。ここでは、「窒化プロセス」という材料科学や製造分野の専門用語が現れているため、“deformation” が適切である。Vanilla と TR-ft-shared の翻訳文はこの「変形」に当たる単語のみしか違いがない。そして、どちらも「窒化」という単語が翻訳できていないことから、モデルが材料科学のドメインの文であることを十分に考慮せずに翻訳したと考えられる。しかし、ここで疑問なのは、“variant” の方が頻度が高いのにもかかわらず、Vanilla は “deformation” を選択した一方で、TR-ft-shared は「変形」の語義を正しく識別できていたの

にもかかわらず、“variant” を選択したことである。現時点では、この原因を特定に至っておらず、今後の課題である。

3. 対象単語: 上回る

   正解の訳語: exceed

   日文: 既存 客 や 見 込 客 に 向け て 、 新た な 価値 を 創造 する ため に 要する コスト は 、 どの よう な 場合 、 期待 収益 を **上回る** の でしょう か 。

   Ref: When does the cost of creating new value for current and targeted customers **exceed** the expected benefits ?

   Vanilla: In what cases will the cost of creating new value for existing and prospective customers **exceed** expectations ? [BLEU:38.1]

   TR-ft-shared: In what cases are the costs required to create new value for existing and prospective clients ? [BLEU:9.5]

   考察:

   Vanilla では “exceed” と適切に翻訳されているが、TR-ft-shared では訳抜けが発生している。提案手法が WSD を導入し、単語を語義毎のトークンに適切に変換し、訳語選択曖昧語に関する情報を翻訳モデルに適切に入力できたとしても、ある程度の訳抜けを防ぐことができるとは考えられるが、完全に防ぐことはできないということを示している。

第5章　おわりに

5.1　本論文のまとめ

本研究では、訳語選択曖昧語を対象に語義曖昧性解消を行い、語義毎にトークンを分けてから翻訳モデルを学習することで、翻訳モデルが多義語を適切な訳語に翻訳する能力を向上させる手法を提案した。まず、パラレルコーパスに対して単語アライメントを実施し、語義毎に目標言語の別の単語に翻訳される訳語選択曖昧語を選定した。次に、訳語選択曖昧語に対する WSD モデルを構築した。訳語選択曖昧語辞書における訳語を語義とし、パラレルコーパスにおける単語アライメントの結果から正解の語義がラベル付けされたデータセットを構築し、BERT をベースとした 3 つの WSD モデルを学習した。さらに、unknown クラスと uncertain クラスを導入し、訳語選択曖昧語辞書外の語義を推測できるような工夫を行った。最後に、パラレルコーパスにおける原言語の文に対し、学習した WSD モデルを用いて訳語選択曖昧語の語義を推定し、語義の ID をトークンに付与することで語義毎にトークンを分け、この処理を行ったパラレルコーパスから Transformer により翻訳モデルを学習した。

実験の結果、提案手法は語義によってトークンを分けないベースラインの翻訳モデルと比べて、BLEU スコアで 0.2 ポイント、訳語選択正解率で約 1 ポイントの向上が見られた。また、訳語選択曖昧語の語義を正しく推定する理想的な翻訳モデルは、BLEU スコアで 1 ポイント、訳語選択正解率で約 16.6 ポイントの大幅な向上が見られ、提案手法の潜在的な有効性が示された。一方で、性能向上を期待して導入した unknown クラスおよび uncertain クラスは、かえって性能低下を招く結果となった。ケーススタディおよびエラー分析による定性的評価では、提案手法によってドメイン知識を反映した翻訳や、訳抜けの抑制、微細なニュアンスの翻訳が可能になることを確認した。その一方で、高度なドメイン知識への対応や、適切な WSD タグが付与されているにもかかわらず低頻度の語義へ誤訳する現象の解明が今後の課題として残された。

5.2　今後の課題

今後の課題として以下の 3 点を挙げる。

1. **WSD モデルの性能向上**

実験結果により、正しい語義タグが付与された TR-gold は大幅な性能向上が見られ、提案手法のアプローチの妥当性が確認された。したがって、WSD モデルの性能を更に向上させることが翻訳モデルの性能を向上させるための方向性として有望である。一方で unknown クラスや uncertain クラスの導入は期待された成果をあげることができなかった。今後は、これらのクラスの定義や推定方法を見直すことで、事前に定義された語義 (訳語) 以外に翻訳されるケースに正しく対応する必要がある。

2. **訳語選択曖昧語辞書の改善**

   本研究で扱った訳語選択曖昧語辞書は GIZA++の出力に基づく単語単位のアライメントを前提としている。しかし、GIZA++は単語の 1 対 1 の対応関係を推定するため、複数の単語で一つの意味を成す熟語への対応が不十分である。例えば、現在の訳語選択曖昧語辞書は、「基づく」という語に対して、“base” と “on” がそれぞれ別の訳語候補として登録されている。そのため、本来の意味で曖昧語ではない単語が訳語選択曖昧語として登録されていたり、訳語選択正解率での適切な評価ができていなかったりする。したがって、今後は熟語やフレーズ単位での対応関係を考慮した、より実用的かつ高精度な辞書の構築が必要である。

3. **他言語・他ドメインへの拡張**

   本研究では日英翻訳での実験を行ったが、異なる言語ペア (例、日中、日独、日西) や、逆方向での翻訳 (英日) における検証が必要である。また、定性分析において、医療や法律といった高度な専門知識を要するドメインでは、適切な訳語が選択されない事例が確認された。曖昧語の誤訳が重大な影響を及ぼすこれらの分野において翻訳モデルの実用性を高めるためには、ドメイン特化コーパスを用いたドメイン適応を行い、特定分野における提案手法の有効性を検証する必要がある。

参考文献

[1] Annette Rios Gonzales, Laura Mascarell, and Rico Sennrich. Improving word sense disambiguation in neural machine translation with sense embeddings. In *Proceedings of the Second Conference on Machine Translation*, p. 11–19, 2017.

[2] Quang-Phuoc Nguyen, Anh-Dung Vo, Joon-Choul Shin, Phuoc Tran, and Cheol-Young Ock. Korean-vietnamese neural machine translation system with korean morphological analysis and word sense disambiguation. *Journal of IEEE Access*, Vol. 7, pp. 32602–32616, 2019.

[3] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of Advances in neural information processing systems*, pp. 3104–3112, 2014.

[4] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1724–1734, 2014.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 2015 International Conference on Learning Representations (ICLR)*, 2015.

[6] Elijah Rippeth, Marine Carpuat, Kevin Duh, and Matt Post. Improving word sense disambiguation in neural machine translation with salient document context. arXiv preprint arXiv:2311.15507, 2023.

[7] Pierpaolo Basile, Lucia Siciliani, Elio Musacchio, and Giovanni Semeraro. Exploring the word sense disambiguation capabilities of large language models. arXiv preprint arXiv:2503.08662, 2025.

[8] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, p. 28–39, 2017.

[9] Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. Integrating weakly supervised word sense disambiguation into neural machine translation. *Journal of Transactions of the Association for Computational Linguistics (TACL)*, Vol. 6, p. 635–649, 2018.

[10] Yang Zhao, Lu Xiang, Junnan Zhu, Jiajun Zhang, Yu Zhou, and Chengqing Zong. Knowledge graph enhanced neural machine translation via multi-task learning on sub-entity granularity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4495—-4505, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics.

[11] Ayush Maheshwari, Preethi Jyothi, and Ganesh Ramakrishnan. Dictdis: Dictionary constrained disambiguation for improved nmt. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10991–11004, Miami, Florida, USA, 2024. Association for Computational Linguistics.

[12] Tomoyuki Kajiwara. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6047–6052, Florence, Italy, 2019. Association for Computational Linguistics.

[13] J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. Parabank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, pp. 6521–6528. Association for the Advancement of Artificial Intelligence, 2019.

[14] Josef Jon, Dusan Varis, Michal Novák, João Paulo Aires, and Ondřej Bojar. Negative lexical constraints in neural machine translation. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pp. 372–384, Macau SAR, China, 2023. Asia-Pacific Association for Machine Translation.

[15] Jinpeng Zhang, Nini Xiao, Ke Wang, Chuanqi Dong, Xiangyu Duan, Yuqi Zhang, and Min Zhang. Disambiguated lexically constrained neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10583–10596, Toronto, Canada, 2023. Association for Computational Linguistics.

[16] Van-Hien Tran, Raj Dabre, Hour Kaing, Haiyue Song, Hideki Tanaka, and Masao Utiyama. Exploiting word sense disambiguation in large language models for machine translation. In *Proceedings of the First Workshop on Language*

Models for Low-Resource Languages, pp. 135–144, Abu Dhabi, United Arab Emirates, 2025. Association for Computational Linguistics.

[17] Jan Niehues and Eunah Cho. Exploiting linguistic resources for neural machine translation using multi-task learning. In *Proceedings of the Second Conference on Machine Translation*, pp. 80–89, Copenhagen, Denmark, 2017. Association for Computational Linguistics.

[18] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Journal of Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, 2003.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008, 2017.

[20] Morishita Makoto, Chousa Katsuki, Suzuki Jun, and Nagata Masaaki. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 6704–6710, Marseille, France, 2022. European Language Resources Association.

[21] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, p. 230–237, Barcelona, Spain, 2004. Association for Computational Linguistics.

[22] 伝康晴, 小木曽智信, 小椋秀樹, 山田篤, 峯松信明, 内元清貴, 小磯花絵. コーパス日本語学のための言語資源：形態素解析用電子化辞書の開発とその応用. 日本語科学, Vol. 22, pp. 101—-123, 2007.

[23] Bird Steven, Loper Edward, and Klein Ewan. *Natural Language Processing with Python*. O'Reilly Media, Inc., 2009. Accessed: 2026-02-18.

[24] 新納浩幸, 森信介, 古宮嘉那子, 佐々木稔. Kytea を利用した日本語 all-words wsd. 言語処理学会第 22 回年次大会, pp. 633–636, 2016.

[25] Tohoku NLP Group. bert-base-japanese-v3. `https://huggingface.co/tohoku-nlp/bert-base-japanese-v3`, 2023. Accessed: 2025-12-13.

[26] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pp. 67–72, 2017.

[27] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311—-318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics.

付 録A　実験結果の補遺

全訳語選択曖昧語もしくはWSDモデルを学習した訳語選択曖昧語のみを対象とした訳語選択正解率のみを示したグラフを図A.1とA.2に示す。Vanillaと TR-ft-sharedの文毎の訳語選択曖昧語の翻訳結果を比較した詳細な対応表をA.3に示す。

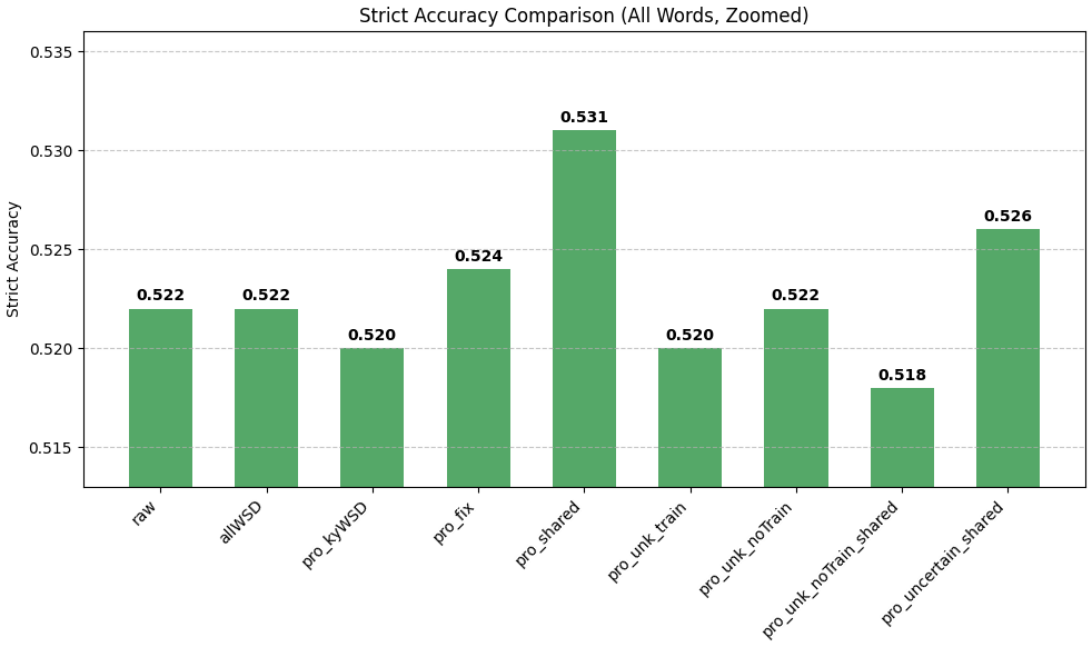


図 A.1: 全訳語選択曖昧語語を対象とした訳語選択正解率

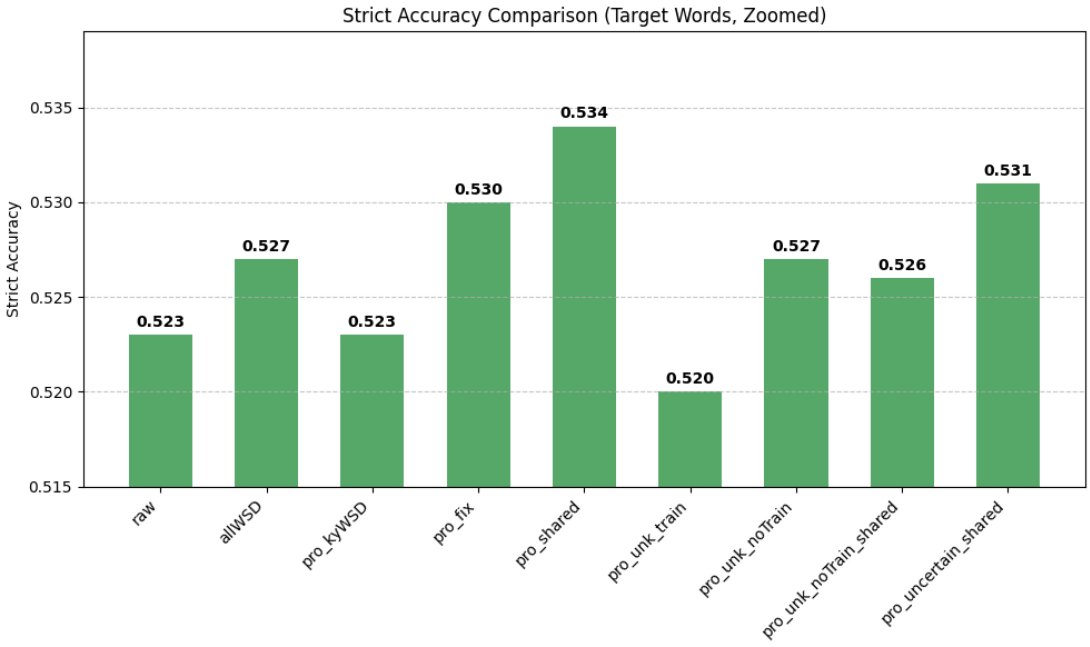
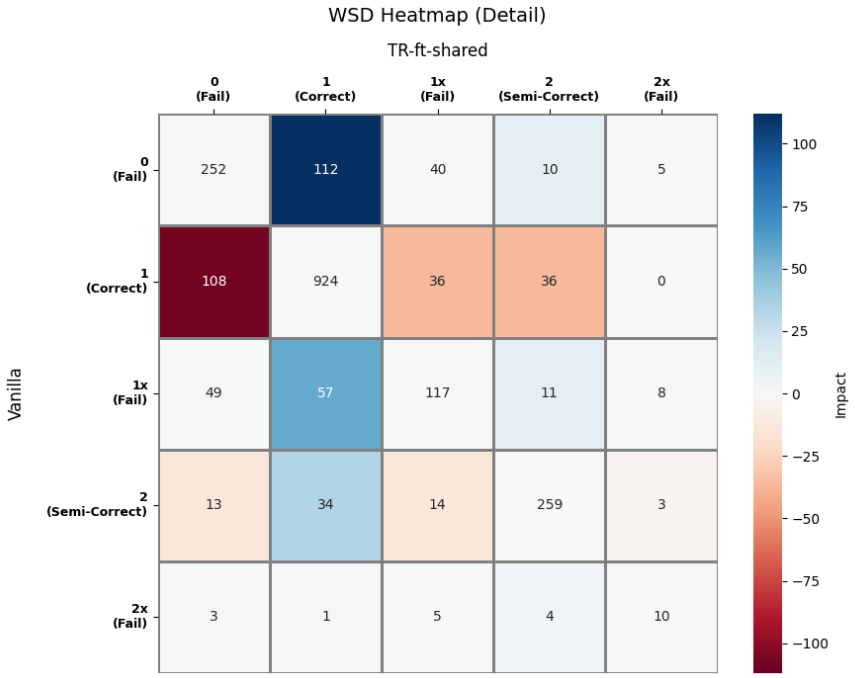


図 A.2: 学習した訳語選択曖昧語のみを対象とした訳語選択正解率



0 は訳語選択曖昧語辞書外の語義を訳出した不正解、1 は正解の語義を訳出した正解、1x は訳語選択曖昧語辞書内の語義 1 語を訳出した不正解、2 は正解の語義と訳語選択曖昧語辞書内の別の語義を訳出した準正解、2x は訳語選択曖昧語辞書内の語義 2 つを訳出した不正解を表す。

図 A.3: Vanilla と TR-ft-shared の訳語選択曖昧語に対する訳語の正解判定結果の対応表 (詳細版)