修士論文

誤りタイプを考慮した日本語文法誤り訂正モデルの学習

SHI Haoda

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 7 年 3 月

Abstract

Grammatical Error Correction (GEC) is a method that automatically corrects grammatical errors in texts written by language learners, which is widely applied for various fields such as language education. While research on GEC has primarily focused on English, studies on Japanese GEC remain limited. In addition, many previous studies have trained a single GEC model without considering error types, thus failing to fully account for the distinct characteristics of each type of error. Furthermore, it is crucial not only to correct grammatical errors but also to provide feedback on the nature of the errors in order to support language learners in understanding their own errors and acquiring correct grammar. However, such a research direction is not well studied in the field of Japanese GEC.

This study proposes a method for Japanese GEC that integrates an error type classification model and individual GEC models that are specialized for each of the error types. First, error types are predefined, and specialized GEC models are trained for each of the error types. For a given erroneous sentence, its type of error is identified. Then, the GEC model corresponding to the identified error type is applied to generate a corrected sentence. It is expected to improve the performance of GEC by utilizing specialized GEC models tailored to each error type. Finally, both the corrected sentence and the identified error type are presented to a learner. Showing the error type helps learners understand the cause of their error, facilitating more effective language learning.

The details of the proposed method are explained as follows. First, seven error types are defined: "postposition," "auxiliary verb," "typographical error," "verb conjugation," "incorrect choice of verb," "incorrect choice of noun," and "other." Pairs of erroneous sentences written by Japanese learners and sentences corrected by native Japanese speakers are extracted from the Lang-8 corpus, an existing dataset of GEC, with preprocessing of removal of edit markers and so on. Next, error types are assigned to each pair of erroneous and corrected sentences by the following procedures: detecting differences between an erroneous sentence and a corrected sentence, performing morphological analysis for word segmentation and part-of-speech tagging, and identifying an error type by manually designed rules. A pair of sentences is removed if an erroneous sentence contains multiple errors. Through these procedures, a dataset consisting of triplets of an erroneous sentence, a corrected sentence, and a label of the error type is constructed, which is then referred to as the "Type-labeled GEC dataset."

Next, the error type of an erroneous sentence is classified. Pre-trained learning

models, specifically BERT and RoBERTa, are used as the base model. They are fine-tuned using the Type-labeled GEC dataset to train error type classification models. Furthermore, data augmentation is applied to increase the number of samples for "typographical error" and "incorrect choice of noun," as there are a few samples of these error types in the dataset. Grammatically correct Japanese sentences are converted to erroneous sentences by artificially introducing errors of "typographical error" or "incorrect choice of noun", where the KeiCo corpus serves as the source for grammatically correct sentences. These synthesized erroneous sentences are then coupled with their original sentences to create additional error correction samples. The generated pseudo-samples are combined with the original dataset, and the error type classification model is fine-tuned using this expanded training data.

Next, the method to revise an erroneous sentence into a grammatically correct sentence is implemented as follows. A sequence-to-sequence model is trained as a GEC model where an erroneous sentence is an input and a corrected sentence is an output. The pre-trained language model T5, which is applicable for sequence-to-sequence tasks, is fine-tuned using the Type-labeled GEC dataset. The dataset is divided into several portions by the error type, and a separate GEC model that focuses on one error type is trained using each portion of the dataset. As a result, seven different GEC models that are particularly tuned on the correction of the specific error type are obtained.

We report the experiments to evaluate our proposed method as follows. The error type classification models are evaluated on two tasks: the error classification task as well as the error detection and classification task. The former is a task to classify an error type of a given erroneous sentence into seven classes (error types). The latter is a task to identify whether a given sentence contains a grammatical error and classify its error type if there is an error, implemented by adding "no error" as a classification class. In the second task, grammatically correct sentences extracted from the Lang-8 corpus are labeled with the "no error" tag and added to the dataset. A dataset consisting of 150 erroneous sentences manually labeled with their error types is prepared and used as test data to evaluate the performance of the error type classification models.

In the error classification task, a comparison between the two language models used as the base model for error type classification showed that RoBERTa achieved a higher F1 score than BERT. The data augmentation did not improve the F1 score for BERT, but it led to an improvement for RoBERTa. The highest F1 score was

0.60, which was obtained by the RoBERTa model with data augmentation. On the other hand, in the error detection and classification task, BERT outperformed RoBERTa, achieving an F1 score of 0.55.

Next, the performance of the GEC models is evaluated. A single GEC model that corrects errors without taking error types into account is used as a baseline, and its performance is compared with the proposed method where individual GEC models for each of the error types are employed. Two variations of the proposed method are considered: $\mathbf{PRO_{gold}}$, which switches GEC models based on the ground-truth error type in the dataset, and $\mathbf{PRO_{auto}}$, which switches the models based on the automatically classified error type. In the $\mathbf{PRO_{auto}}$, the error type classification model is the RoBERTa model that has been fine-tuned with the augmented dataset. The GLEU score is used as the evaluation criterion for GEC. $\mathbf{PRO_{gold}}$ outperformed the baseline in all error types except for "postposition." The overall GLEU score of $\mathbf{PRO_{gold}}$ (0.7739) was higher than that of the baseline (0.7618). However, the GLEU score of $\mathbf{PRO_{auto}}$ was 0.7390, which was lower than the baseline. Especially, a significant decline in performance was observed for the "auxiliary verb" error type. This may be caused by the low performance of error type classification. The improvement of the performance of the error type classification model remains important future work.

概要

文法誤り訂正は語学学習者が作成したテキスト内の文法的な誤りを自動的に修正する技術であり、語学教育など幅広い分野で活用されている。これまで主に英語を対象として研究が発展してきたが、日本語の文法誤り訂正の研究はまだ十分に進んでいない。また、先行研究の多くは誤りのタイプを考慮せずに単一の文法誤り訂正モデルを学習し、誤りタイプ毎に異なる誤りの特徴を十分に考慮できていないという課題がある。さらに、学習者が自身の誤りを理解し正しい文法を修得するには、単に誤りを訂正するだけでなく、どのような誤りであるかをフィードバックすることも重要であるが、日本語の文法誤り訂正を対象とした研究は進んでいない。

本研究では、誤りタイプ分類モデルと個々の誤りタイプに特化した誤り訂正モデルを組み合わせた日本語文法誤り訂正手法を提案する。まず、誤りのタイプをあらかじめ定義し、それぞれの誤りのタイプに特化した誤り訂正モデルを個別に学習する。誤りのある文が入力されたとき、その誤りのタイプを分類する。次に、分類した誤りのタイプに応じて、それに対応する誤り訂正モデルを用いて文法的に正しい文を生成する。誤りタイプ毎にその特徴を反映した誤り訂正モデルを使い分けることで誤り訂正の性能向上を狙う。最後に、学習者に訂正文と誤りのタイプを提示する。誤りタイプの提示により学習者に誤りの原因を理解することを促す。

提案手法の詳細を以下に述べる。まず、誤りタイプを「助詞」「助動詞」「表記」「動詞の活用」「動詞 (単語選択)」「名詞 (単語選択)」「その他」と定義する。既存の誤り訂正のデータセットである Lang-8 コーパスから、不要な記号の除去などの前処理を行い、日本語学習者の誤り文と日本語ネイティブスピーカーによって誤りを修正した文の組の集合を収集する。次に、誤り文と訂正文の差分の検出、形態素解析による単語分割と品詞付け、および人手で設計したルールによって、誤り文と訂正文の組に対して誤りタイプのラベルを付与する。このとき、誤りを 2 つ以上含む文は除外する。以上の手続きで、誤り文、訂正文、誤りラベルの 3 つ組から構成される「タイプ付き誤り訂正データセット」を構築する。

次に、誤りのタイプを分類する。事前学習済み学習モデル BERT および RoBERTa を基盤モデルとし、タイプ付き誤り訂正データセットを用いてこれをファインチューニングすることで、誤りタイプを分類するモデルを学習する。さらに、データセットにおいてサンプル数が少ない「表記」「名詞 (単語選択)」の誤りタイプについて、誤り事例の数を増やすためのデータ拡張を行う。KeiCO コーパスを利用し、文法的に正しい日本語文に「表記」や「名詞 (単語選択)」の誤りをルールベースの手法で人為的に発生させ、これを元の文と組み合わせることで誤り訂正のサンプルを生成する。生成した擬似サンプルと元のデータを結合し、これを訓練データと

して誤りタイプ分類モデルをファインチューニングする。

次に、誤りを含む文を文法的に正しい文に訂正する方法について述べる。誤り訂正モデルとして、誤り文を入力、訂正文を出力とする系列変換モデルを学習する。系列変換に適用できる事前学習済み言語モデルである T5 を用い、タイプ付き誤り事例データセットを用いてこれをファインチューニングする。この際、データセットを誤りタイプによって分割し、誤りタイプ毎に個別の誤り訂正モデルを学習する。結果として 7 種類の誤り訂正モデルを得る。

提案手法の評価実験について述べる。誤りタイプ分類モデルの評価実験では、「誤り分類タスク」と「誤り検出・分類タスク」の 2 つによって評価する。誤り分類タスクは、誤りを含む文が 7 種類の誤りタイプのどれに該当するかを分類するタスクである。一方、誤り検出・分類タスクは、「誤りなし」をクラスを追加し、文が誤りを含むか、含むときにはどの誤りタイプに該当するかを判定するタスクである。このタスクでは Lang-8 コーパスから抽出した文法的に正しい文に「誤りなし」のラベルを付与してデータセットに追加する。評価データとして、150 の誤り文に対して人手で誤りタイプを付与したデータを用いる。

誤り分類タスクについて、誤りタイプ分類モデルのベースとした 2 つの言語モデルを比較すると、RoBERTa は BERT と比べて F1 スコアが高かった。また、データ拡張によって、BERT では F1 スコアが改善しなかったが、RoBERTa では改善した。一番良い F1 スコアはデータ拡張ありの RoBERTa モデルで、その値は 0.60 であった。一方、誤り検出・分類タスクについて、BERT は RoBERTa を上回り、その F1 スコアは 0.55 であった。

次に、誤り訂正モデルの評価実験について述べる。誤りタイプを考慮しない単一の誤り訂正モデルをベースラインとし、誤りタイプ毎に個別の訂正モデルを適用する提案手法と比較する。提案手法として、正解の誤りタイプのラベルを用いる $\mathbf{PRO}_{\mathbf{gold}}$ と、誤りタイプを自動判定する $\mathbf{PRO}_{\mathbf{auto}}$ の 2 つを用いる。$\mathbf{PRO}_{\mathbf{auto}}$ では誤りタイプ分類モデルとしてデータ拡張後のデータセットでファインチューニングされた RoBERTa を用いる。誤り訂正の評価基準として GLEU スコアを用いる。$\mathbf{PRO}_{\mathbf{gold}}$ は誤りタイプ「助詞」を除く全てのタイプでベースラインを上回り、全体の GLEU スコア (0.7739) もベースライン (0.7618) より高い結果となった。一方、$\mathbf{PRO}_{\mathbf{auto}}$ の GLEU スコア (0.7390) はベースラインよりも低く、特に「助動詞」の誤りタイプでは大きく性能が低下した。これは誤りタイプ分類モデルの F1 スコアが十分に高くないことが原因として考えられ、誤りタイプ分類モデルの精度向上が今後の課題として残された。

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　背景

文法誤り訂正（Grammatical Error Correction; GEC）は、語学の学習者が作文したテキストに含まれる文法的な誤りを自動的に訂正する技術であり、語学学習の支援、教育支援、オンライン教育、自動評価システムなど、様々な場面で活用されている。この技術は、学習者の学習効率を向上させるだけでなく、教師の負担を軽減し、より効果的な教育環境の構築に寄与する。

文法誤り訂正の研究はこれまで盛んに行われており、特に英語を中心に多くの研究成果がある。一方で、日本語における文法誤り訂正の研究は比較的少なく、その応用範囲はまだ限られている。特に、日本語学習者の誤り傾向に基づく誤り訂正や誤りタイプを提示するシステムの開発は、重要であるにも関わらず、これまで十分な研究が行われていなかった。

また、学習者が文法誤りの本質を理解し、正しい用法を習得するためには、単に文法誤りを訂正するだけでなく、「どのような誤りがあったのか」といった情報を提供することも重要である。誤りのタイプを明示することで、学習者は自身の弱点を把握し、より効率的な学習計画を立てることができる。したがって、誤り訂正と誤りタイプの提示を同時に行うシステムの必要性が高まっている。しかし、現時点では、日本語を対象としたそのようなシステムの研究は盛んではなく、学習者や教育現場からの要望を満たせていない状況にある。

1.2　目的

本研究は、誤りのタイプを十分に考慮した日本語の文法誤り訂正手法を探究することを目的とする。具体的には、文法誤りを含む文が与えられたとき、以下の手順で処理を行う。

まず、文に含まれる誤りのタイプを分類する。ここでの誤りタイプは「助詞の語用」「動詞の活用の誤り」などとする。誤りタイプを自動分類するモデルは、誤りタイプがラベル付けされたデータセットから学習する。次に、誤りを含む文を

文法的に正しい文に変換する。誤り文と正しい文の組からなるデータセットから、誤り文を入力、正しい文を出力とする系列変換モデルを学習し、これを用いて文法誤りを訂正する。さらに、文を訂正する系列変換モデルは誤りのタイプごとに個別に学習するアプローチを採用する。つまり、誤りの種類に特化した訂正モデルを学習し、誤りタイプに応じて使い分けることで、汎用的な GEC モデルよりも高い精度で文法誤りを訂正することを狙う。さらに、本研究の提案システムは学習者に対して修正後のテキストと誤りのタイプの両方を提示する。これにより、学習者に自身の誤りの内容を理解させ、正しい用法を修得することを促す。すなわち、学習者に有効なフィードバックを提供することも視野に入れている。

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、本研究の関連研究について述べる。3 章では、提案手法の詳細を説明する。データセットの構築、誤りタイプの分類、文法誤り訂正モデルの学習などについて述べる。4 章では、提案手法の評価実験について述べる。最後に 5 章では、本研究のまとめと今後の課題について述べる。

第2章　関連研究

本章では本研究の関連研究について述べる。本論文の主な研究トピックは文法誤り訂正 (Grammatical Error Correction; GEC)[2, 3] である。文法誤り訂正は、語学学習支援や教育現場での活用が期待される重要な技術であり、これまで主に英語を対象に数多くの研究が行われてきた。一方で、日本語の文法誤り訂正を対象とし、日本語学習者特有の誤り傾向や誤りタイプの分類に着目した研究はまだ十分とは言えない。また、近年の自然言語処理分野では、事前学習済み言語モデル (Pre-trained Language Models)[6] や系列変換モデル (Sequence-to-Sequence Model)[18] の発展により、文法誤り訂正の精度が飛躍的に向上しているが、これらの技術を日本語の誤り訂正に応用した研究も限定的である。

以下、2.1 節では、文法誤り訂正に関する既存研究について概観し、これらの研究が抱える課題を整理する。また、日本語学習者の誤り傾向に関する研究や誤りタイプ分類の研究を紹介する。2.2 節では、本研究で使用する事前学習済み言語モデルを紹介する。2.3 節では、同じく本研究で使用する系列変換モデルについて説明する。2.4 節では本研究の特色について述べる。

2.1　文法誤りに関する研究

2.1.1　文法誤り訂正

文法誤り訂正は、言語学習者が書いた文章に含まれる文法的な誤りを自動的に訂正する技術である。語学教育、オンライン学習支援、AI を用いた教育ソリューションなど幅広い場面で応用されている。GEC の研究は、初期の統計的手法やルールベースの方法から始まり、近年では機械学習、ニューラルネットワーク、そして事前学習済み言語モデルを活用した手法に進化している。さらに、データセットや評価基準の整備も進み、この分野の研究を支える基盤として確立されている。

初期の GEC 研究はルールベースの手法が主流であった。Dale らは、GEC のタスクを「非母語話者による学術論文に含まれる文法的な誤りやスタイルの不適切さを検出し、それを修正すること」と定義してた上で、事前に定義したルールと

統計情報を用いて文法的な誤りを検出し、それを訂正する手法を提案した [5]。この研究は、標準的な評価指標とデータセットも提供し、初期の GEC 研究の基盤を築いた。

機械学習に基づくアプローチは、GEC 性能の向上に重要な役割を果たした。Rozovskaya らは、誤りの分布をモデリングすることで文法誤りを訂正する手法を提案し、GEC システムの性能を向上させた [17]。ナイーブベイズ分類器と最大エントロピーモデルを用い、文脈に関する特徴量を組み合わせることで、誤りの分布に偏りがあるデータセットに対する機械学習モデルの文法訂正能力を改善させた。Dahlmeier らは Alternating Structure Optimization (ASO) を導入し、複数の誤りタイプを統一的に学習するための構造化学習フレームワークを提案した [4]。ASO により誤りタイプ間の関連性を学習することで、文法誤り訂正タスクにおける修正精度を向上させた。

近年、ニューラルネットワークや系列変換モデルの導入により、GEC の性能が大幅に向上した。Yuan らは初めてニューラル機械翻訳 (Nueral Machine Translation; NMT) 技術を GEC タスクに適用した [20]。Zhao らは、ラベルなしデータを活用したコピー拡張アーキテクチャを提案し、誤り訂正能力を向上させた [21]。

事前学習済み言語モデルの利用により、GEC の精度がさらに向上した。Kaneko らは、BERT(Bidirectional Encoder Representations from Transformers)[6] のような事前学習済みモデルを活用し、エンコーダ・デコーダモデルとして実現した GEC モデルの性能を向上させた [8]。さらに、Lichtarge らは大規模なデータセットを使用して Transformer ベースのモデルを事前学習し、その後 Lang-8[1] のような小規模で高品質なデータセットでファインチューニングを行うことで、文法誤り訂正の性能を大幅に向上させた [10]。

最近の研究では、マルチタスク学習が注目を集めている。Rotman らは、編集操作の予測（挿入、削除、置換などの修正操作を予測）と訂正文の生成（元の文から訂正後の文を生成）という 2 つのタスクを統合的に学習した [16]。このようなマルチタスク学習によって、モデルに文法誤り訂正に必要な多様な「スキル」を習得させ、その訂正能力を向上させた。また、モデルの学習時における複数の訓練データセットの使用順序や各訓練データ内のインスタンスの配置が最終的な性能に重要な影響を与えることを発見し、最適なトレーニングスケジュールを設定することで、より小さなモデルで最先端の性能を達成した。

GEC の評価基準の整備も GEC 研究の発展において重要な役割を果たしている。例えば、Napoles らは GLEU などの GEC の自動評価指標を提案し、GEC システムの評価の標準化を促進した [14]。

[1] Lang-8 の詳細は 3.3.1 項で述べる。

2.1.2　日本語学習者の誤り傾向

水本らは、語学学習 SNS「Lang-8」の添削ログを活用し、日本語学習者が犯す誤りの種類やその傾向について詳細に分析した [13]。ここでは発生の頻度が高い「助詞の誤り」、「語彙選択の誤り」、「表記の誤り」の 3 種類について説明する。

助詞は日本語の文法において重要な役割を果たしているが、学習者にとってその使い分けは難しい。水本らの研究によると、助詞の誤りは全体の 25%を占め、最も頻繁に見られる誤りであった。特に、「は」と「が」、「を」と「に」の混同が典型的な例として挙げられる。以下は助詞の誤りの例である。

(例)
誤：私は図書館に行きます。
正：私は図書館へ行きます。

このような誤りの原因として、学習者の母語と日本語の文法の違いが挙げられる。例えば、英語や中国語のように助詞に相当する文法要素が存在しない、もしくは機能が異なる言語を母語とする学習者にとって、日本語特有の助詞の使い方を正確に理解することは難しい。また、助詞は文脈に応じて使い方が変わるため、その使い分けの学習も難しい。

語彙選択の誤りは、学習者が文脈や場面に適した単語を選択できなかった誤りである。このタイプの誤りは全体の 15%を占める。類義語や似た意味を持つ単語（例：「大変」と「難しい」）を混同して使用するケースが特に多い。例を以下に挙げる。

(例)
誤：この問題は大変です。
正：この問題は難しいです。

語彙選択の誤りは、学習者が単語の意味のニュアンスを十分によく理解していなかったり、使用場面に応じた単語の適切な選択ができていなかったりすることから生じる。さらに、学習者が単語の意味のような辞書的な知識を持っていても、その単語を使うのに適した文脈に関する知識を持っていないことによっても生じる。特に中級レベル以下の学習者に多く見られる傾向がある。

表記の誤りは、綴りの間違いであり、日本語特有の漢字、ひらがな、カタカナの表記体系に起因する。この誤りは全体の 10%を占める。特に、漢字の誤字や不適切な使用、ひらがなとカタカナの混同が典型的な誤りである。例を以下に挙げる。

(例)
誤：昨夜、食道しました。(「食事」が正しい)
正：昨夜、食事しました。

表記の誤りは、学習者の漢字に対する習熟度が不十分であることや、学習者が日常生活で漢字を頻繁に使用しない環境にいることが原因であると考えられる。また、形状が似ている漢字を誤って書く場合も見られる。

2.1.3 日本語の文法誤り訂正

日本語の文法誤り訂正は、日本語学習者が作成した文章に含まれる文法的な誤りを自動的に検出し、文法的に正しい文章に訂正する技術である。日本語の GEC は英語 GEC の研究ほど進展していないものの、近年注目を集めており、多くの有望なアプローチが提案されている。

水本らは、語学学習 SNS「Lang-8」のデータを基に、文法誤り訂正システムの構築に有用なコーパスを構築した [13]。さらに、日本語母語話者による訂正ログを基に、学習者の誤用パターンを体系的に分類し、その結果を基に文法誤り訂正システムを構築した。特に、助詞や語彙選択に関する誤りの自動訂正に注力した。

甫立らは、日本語の文法誤り訂正において多様な訂正文を生成するアプローチを提案した [7]。この手法では、ニューラルネットワークを用い、入力文に対して複数の訂正候補を生成し、それらを比較して最適な訂正文を選択する。この方法により、文脈に応じた柔軟な訂正が可能となった。

新井らは、日本語学習者向けの文法誤り検出機能付き作文用例検索システムを提案した [1]。本システムは、双方向 LSTM(Bidirectional Long Short-Term Memory; BiLSTM) を用いて文法誤りを高精度に検出し、適切な修正結果と、学習者の文法誤りに対する正しい用例を学習者に提示する機能を備えている。従来の用例検索システムでは、誤りを含む文 (クエリ) に対して適切な用例を検索できない問題があったが、文法誤り検出機能を組み込むことでこの課題を解決している。評価実験では、提案システムが用例検索精度の向上と学習者の作文品質の改善に寄与することが確認された。本システムは、学習者が自身の誤りを認識し、正しい日本語表現を学ぶための有効な支援ツールとなる可能性がある。

2.1.4 文法誤りのタイプ分け

文法誤り訂正において、誤りのタイプ分けはシステムの性能評価や改善において重要な役割を果たす。英語の GEC 分野では、ERRANT(Error Annotation Toolkit)

が誤りタイプの分類に広く用いられており、その効率性と汎用性が評価されている [2]。ERRANT は、文に文法誤りのタイプをアノテーションするツールキットとしても利用される。

ERRANT の主な特徴は以下の通りである。第一に、言語非依存性が挙げられる。ERRANT は特定のデータセットや言語に依存せず、事前の訓練データやアノテーションデータが不要であるため、さまざまな言語の GEC システムに応用可能である。第二に、ルールベースの手法を採用しており、エラータイプを明確かつ一貫性のある方法で分類できる。

ERRANT は、日本語 GEC 研究にも影響を与えている。小山らは、英語版 ERRANT を参考に日本語に特化した誤用タグの設計を行い、誤用タグ付き評価コーパスを構築した [9]。日本語における文法誤りを体系的に分類する手法を提案し、語学学習 SNS「Lang-8」の日本語学習者コーパスに対して語用タグを付与した。この語用タグ付きコーパスは、日本語 GEC モデルの評価や異なるモデル間の性能比較に利用できる。

2.2 事前学習済み言語モデル

事前学習済み言語モデルは、文の意味を理解する処理の基盤となるモデルであり、あらかじめ大量のテキストから事前に学習される。事前学習済み言語モデルは自然言語処理において大きな進展をもたらしたアプローチである。特に BERT[6] と RoBERTa(A Robustly Optimized BERT Pretraining Approach)[12] は、代表的なモデルとして広く使用されている。本節では、これらのモデルの原理と特徴について述べる。

BERT は、双方向の Transformer[19] を基盤としている。事前学習時に双方向 (文頭から文末、もしくは文末から文頭) の文脈情報を同時に捉えることで、高度な意味理解を可能にしている。BERT の事前学習には以下の 2 つのタスクが用いられる。1 つ目のタスクは Masked Language Model (MLM) である。入力文中の一部のトークンをマスクし、それを予測するタスクを通じて、文脈情報を学習する。モデル学習時の損失関数は以下のように定義される。

$$\mathcal{L}_{\mathrm{MLM}} = -\sum_{i=1}^{N} \log P(t_i|\mathbf{T}_{\setminus i}) \tag{2.1}$$

ここで、t_i はマスクされた単語、$\mathbf{T}_{\setminus i}$ はマスクされた単語以外の単語の集合 (文) である。2 つ目のタスクは Next Sentence Prediction (NSP) である。2 つの文が連続するか否かを判定することで、文間の関係性を学習する。このタスクの損失関

数は以下のように定義される。

$$\mathcal{L}_{\mathrm{NSP}} = -\big[y \log P_{\mathrm{NSP}} + (1-y)\log(1-P_{\mathrm{NSP}})\big], \tag{2.2}$$

ここで、y は文の連続性ラベル、すなわち 2 つの文が連続する (1) か否か (0) を表すラベルである。

RoBERTa は、BERT の事前学習プロセスを改良した事前学習済み言語モデルである。RoBERTa では、事前学習時にいくつかの重要な変更が加えられている。第一に、動的マスキングを採用し、各エポックにおいて文の異なる部分をマスクすることで、より多様なデータから言語モデルを学習する。第二に、BERT で用いられていた NSP タスクを廃止し、連続した複数の文からなる長い単語列を使用してモデルを学習する。第三に、事前学習のために BERT よりも大規模なテキストデータを使用し、またより長い時間をかけて言語モデルを学習している。これらの改良により、RoBERTa は BERT を超える性能を示しており、特に複雑な文脈理解を必要とするタスクにおいて優れた成果を挙げている。

2.3 系列変換モデル

系列変換モデル (Sequence-to-Sequence Model または Seq2Seq Model) は、入力された系列データを別の系列データに変換するモデルであり、機械翻訳、要約、GEC など、幅広い自然言語処理タスクで利用されている。

典型的な系列変換モデルは、RNN(Reccurent Neural Network) に基づいたエンコーダー・デコーダー構造を持つ。このモデルでは、エンコーダーが入力系列を処理し、その入力系列の特徴を表す「コンテキストベクトル」に変換し、デコーダーがそれを基に出力系列を生成する。文法誤り訂正タスクは、入力文を「誤りのある文」、出力文を「正しい文」とする系列変換問題として定式化され、系列変換モデルを学習することで解くことができる。このとき、誤りのある文中の文脈情報を正確に捉えるモデルを学習することが重要である。

しかし、従来の系列変換モデルには、(1) 長い系列における単語間の依存関係を正確に捉えられない、(2) エンコーダーが生成する固定長のコンテキストベクトルが情報損失を引き起こす可能性がある、といった課題が存在する。これらの課題を解決するために注意機構 (Attention Mechanism または Attention)」が考案された。Attention は、モデルが入力系列全体の情報を考慮できるよう、エンコーダーの各隠れ状態 $\mathbf{h}_i$ を重み付けしてコンテキストベクトルを生成する。この重みは以下のように計算される。

$$\alpha_{ij} = \frac{\exp(\mathbf{h}_i^\top \mathbf{s}_j)}{\sum_k \exp(\mathbf{h}_k^\top \mathbf{s}_j)} \tag{2.3}$$

ここで、α_{ij} はエンコーダーの隠れ状態 $\mathbf{h}_i$ とデコーダーの隠れ状態 $\mathbf{s}_j$ の間の注意スコアを正規化したものである。α_{ij} による重み付けにより、デコーダーは入力系列全体から文脈情報を抽出することができる。Attention により、系列変換モデルは長い依存関係を持つ系列データを適切に扱うことができるようになった。

T5(Text-to-Text Transfer Transformer) は、Google Research によって提案された系列変換モデルである [15]。すべての自然言語処理タスクを「テキストからテキストへの変換」として定式化するアプローチを採用している。この統一的なアプローチにより、複数のタスクを単一のモデルで扱うことが可能である。T5 の特徴として以下が挙げられる。

タスク形式の統一 T5 はすべてのタスクをテキスト形式で扱うため、分類タスクや生成タスクなど異なるタスクを統一的に処理できる。

エンコーダー・デコーダー構造 T5 は Transformer[19] に基づいたエンコーダー・デコーダー構造を採用しており、入力文の長距離依存関係を正確に捉える能力を持つ。

事前学習とファインチューニング BERT や RoBERTa と同様に、T5 は大規模なコーパスで事前学習を行い、その後特定のタスクにファインチューニングすることで、様々な下流タスクに柔軟に適用できる。

T5 は GEC にも自然に適用できる。すなわち、T5 の入力を「誤りのある文」、出力を「正しい文」と定義すればよい。また、事前学習済み T5 モデルを GEC のデータセットを用いてファインチューニングすることで、優れた GEC モデルを学習できる。

2.4 本研究の特色

本研究の特色は、日本語の文法誤り訂正タスクにおいて、誤りタイプ分けと誤り訂正モデルを組み合わせた新しいアプローチを採用している点にある。従来の GEC モデルは、すべての誤りタイプを単一のモデルで処理するため、誤りタイプごとの特性を十分に考慮できないという課題があった。この課題を克服するため、「助詞」「助動詞」「表記」など 7 種類の誤りタイプを定義し、これらの誤りタイプ毎に個別の誤り訂正モデルを学習する。誤り文が入力されたとき、誤りタイプを推定し、その誤りタイプに特化した誤り訂正モデルを用いて訂正文 (文法的に正しい文) を生成する。個々の誤り訂正モデルは誤りタイプ毎の特徴を反映しているため、これらを誤りタイプに応じて使い分けることで文法誤り訂正の性能の向上が期待できる。

第3章　提案手法

3.1　概要

本研究では、日本語学習者が作成した文に含まれる文法的な誤りを自動的に分類した上で、その誤りを訂正し、文法的に正しい文に変換するシステムを提案する。

提案手法の概要を図 3.1 に示す。提案手法は大きく 2 つのステップに分けられる。最初のステップは誤りタイプの分類である。図 3.1 中の E_i は誤りのタイプを表す。例えば、「助詞の語用」「動詞の活用の誤り」などを誤りタイプとする。第 2 のステップは誤り訂正である。文法誤りを含む文を文法的に正しい文に変換するモデルを用いる。ただし、文法誤り訂正モデルは誤りタイプ毎に個別に学習する。図 3.1 中の GEC_i は誤りのタイプ E_i に特化した文法誤り訂正モデルを表す。この 2 つのステップを経て、誤りが訂正された文と誤りのタイプが学習者に提示される。

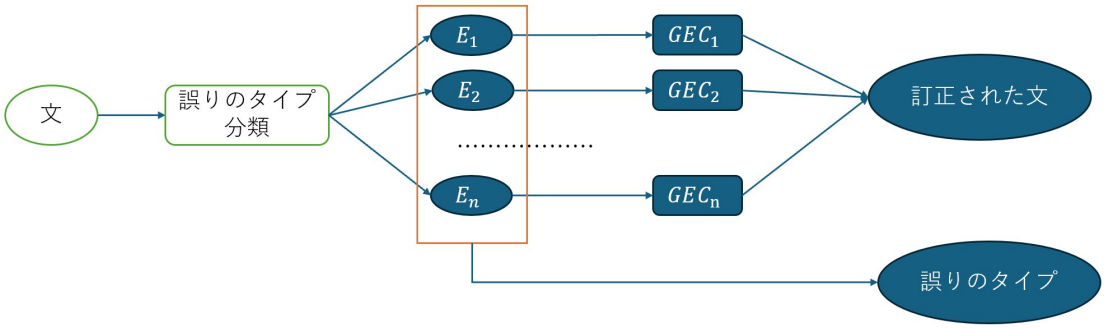

図 3.1: 提案手法の概要

従来の GEC の手法では、誤りのタイプに依らず、誤り文を正しい文に変換する単一のモデルが学習されていた。しかし、日本語学習者による誤りの発生の原因は、文の内容だけではなく、学習者の母語の影響や学習者の学習進度など様々な要因が複雑に関係している。そのため、既存の GEC モデルでは誤り発生の原因を十分に解析できず、そのために文法誤りを正確に訂正できない可能性がある。これに対し、本研究では誤りタイプ毎に GEC モデルを学習することで、誤りの要因

が複雑に絡み合うことを抑制し、その結果誤り訂正の性能が向上することが期待できる。これに加え、語学学習のための有益なフィードバッグを学習者に与えるために、単に誤りを訂正するだけでなく、誤りのタイプも学習者に提示する。

以下、2 つのステップの詳細な問題設定について述べる。

誤りタイプの分類　文法誤り分類タスクは式 (3.1) のように定式化される。

$$\hat{y} = \arg\max_{y \in \mathcal{C}} P(y|\mathbf{x}) \tag{3.1}$$

ここで、$\mathbf{x}$ は入力文、y は誤りタイプ、$\mathcal{C}$ は誤りタイプの集合、$P(y|\mathbf{x})$ は $\mathbf{x}$ の文法誤りが y に該当する確率を表す。誤りタイプ分類モデルは確率 $P(y|\mathbf{x})$ を与えるモデルである。本研究では事前学習済み言語モデルをファインチューニングすることで誤りタイプ分類モデルを作成する。

誤り訂正　本研究では、入力された誤り文を正しい文に変換する系列変換モデルを学習することで誤り訂正を実現する。誤り訂正タスクは式 (3.2) のように定式化される。

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \tag{3.2}$$

ここで、$\mathbf{x}$ は誤り文、$\mathbf{y}$ は訂正後の文、$P(\mathbf{y}|\mathbf{x})$ は $\mathbf{x}$ が $\mathbf{y}$ に訂正される確率である。本研究では、系列変換モデルとして事前学習された言語モデルをファインチューニングすることで得られたモデルによって確率 $P(\mathbf{y}|\mathbf{x})$ を計算する。

3.2　誤りタイプの定義

本節では誤りタイプの定義について述べる。水本らの研究 [13] を参考に、日本語学習者が作文において犯しやすい文法的な誤りを 7 つのタイプに分類する。誤りタイプおよびその例を表 3.1 に示す。

以下、それぞれの誤りタイプについて説明する。

E1. 助詞の誤り　学習者が適切な助詞を使うことができなかった誤りである。学習者が助詞の正しい使い方を理解できていない場合に発生する。助詞は、複数の助詞が似たような役割を果たしたり、文脈に応じて適切なものを使う必要があったりするため、学習者が誤りやすい。特に文の主題を表す助詞「は」と主語を表す助詞「が」の使い分けが難しい。

表 3.1: 誤りのタイプと訂正例

誤りのタイプ	誤りを含む文	正しい文
E1 助詞	家にのんびりしているつもりはないです。	家でのんびりしているつもりはないです。
E2 助動詞	あの時はクロという大きな犬を飼っています。	あの時はクロという大きな犬を飼っていました。
E3 表記	このソフトウェアは最新のテクロジーを取り入れています。	このソフトウェアは最新のテクノロジーを取り入れています。
E4 動詞の活用	最近ちょっと忙しいので、あまり日本語の日記を書きませんでした。	最近ちょっと忙しいので、あまり日本語の日記を書けませんでした。
E5 動詞（単語選択）	三年前に日本にきった。	三年前に日本に来ました。
E6 名詞（単語選択）	本田さんの球が素晴らしい !!	本田さんのボールが素晴らしい !!
E7 その他	その部屋は美しいです。	その部屋はきれいです。

E2. 助動詞　学習者が正しい助動詞を選択できなかった誤りや、助動詞の活用形が正しくなかったという誤りである。助動詞は敬語表現、否定表現、可能表現など様々な役割を持つため、学習者にとってその使い分けが難しい。

E3. 表記　単語綴りの誤りである。日本語が漢字、ひらがな、カタカナという特有の表記体系を持つことに起因する誤りといえる。このタイプの誤りには、漢字の誤字、ひらがなとカタカナの混同などが含まれる。特に、類似した形状を持つ漢字が混同されることが多い。学習者が漢字に慣れていない場合によく生じる誤りである。

E4. 動詞の活用　学習者が動詞の活用形を間違えるケースである。動詞の活用に関する知識や理解が不足していることに起因する。

E5. 動詞 (単語選択)　学習者が文脈や場面に応じた適切な動詞を選択できないといった誤りである。類義語や意味の近い動詞が混同して使われることが多く、動詞の使い分けに関する学習者の理解不足が原因である。

E6. 名詞 (単語選択)　E5 と同様に、学習者が文脈や場面に応じた適した名詞を選択できないといった誤りである。日本語の語彙の豊富さや類似した意味を持つ名詞が多いことに起因する。また、辞書的な意味と実際の使用場面が異なる名詞が存在することもこの誤りの要因のひとつである。

E7. その他　上記のいずれにも分類されない誤りである。接続詞や副詞の不適切な使用、文構造の不自然さ、または文全体の意味的な不整合などが該当する。表 3.1 の例では、「美しい」は対象が人や花などのときに使われる形容詞であるが、対象が部屋のときに誤って使われている。この場合は「きれい」という形容詞を使う方が自然である。

3.3　データセットの構築

本節では、本研究で使用する「タイプ付き誤り訂正データセット」の構築について説明する。このデータセットは、学習者が作成した文法誤りを含む文 (誤り文)、それを文法的に正しい文に訂正した文 (訂正文)、誤り文の誤りタイプ、の 3 つ組を収録したデータセットである。同データセットは、誤り訂正モデルの学習や誤り訂正モデルの学習に用いる。

3.3.1　Lang-8 コーパス

タイプ付き誤り訂正データセットは Lang-8 コーパスをベースに構築する。Lang-8 コーパスは、語学学習者が作成した作文とその添削結果を含む大規模なデータセットであり、主に文法誤り訂正の研究に利用されている。このコーパスは、語学学習者向けの相互添削型ソーシャル・ネットワーキング・サービスである Lang-8 (https://lang-8.com/) から収集されたデータに基づいている。Lang-8 では、語学学習者が作文を投稿し、ネイティブスピーカーや上級学習者がそれを添削するプラットフォームが提供されている。

Lang-8 コーパスは、学習者が投稿したオリジナルの文章、ネイティブスピーカーや上級学習者による訂正後の文章、学習者の母語情報などが含まれている。奈良先端科学技術大学院大学 (NAIST) の松本研究室によって、Lang-8 に 2012 年から

2019 年にかけて投稿された作文と添削結果が収集され、公開されている。なお、Lang-8 コーパスは教育や研究目的に限り利用が許可されており、商用利用を希望する場合は提供元への問い合わせが必要である。Lang-8 コーパスには、英語、日本語、中国語、韓国語、スペイン語、フランス語など、様々な言語のデータが含まれている。特に英語が 1,069,549 件、日本語が 925,588 件と、これら 2 つの言語のデータが最も多く収録されている。

本研究では、Lang-8 コーパスにおける日本語データ使用する。既に述べたように、日本語のエントリ件数は 925,588 と大規模であり、学習者の多様な誤りパターンを網羅していると考えられる。また、誤り訂正モデルを学習するために十分な量が確保されていると言える。

表 3.2: Lang-8 コーパスにおけるデータの例

原文	訂正文	訂正文 (タグ除去後)
こんいちはみなさん！	こん [f-red] に [\f-red] ちはみなさん！	こんにちはみなさん！
きれいホテルをたざいしました。	きれい [f-red] な [\f-red] ホテル [f-red] にたいざい [\f-red] しました。	きれいなホテルにたいざいしました。
私の日本語へたです。	私の日本語 [f-red] は [\f-red] へたです。	私の日本語はへたです。
いちがつに Las Vegas がいきました。	いちがつに [f-blue] ラスベガス [\f-blue][sline][f-red] が [\f-red][\sline][f-red] に [\f-red] いきました。	いちがつにラスベガスにいきました。

Lang-8 コーパスには学習者の文法誤りをネイティブスピーカーや上級学習者が添削した X‘結果が含まれているが、この中には特定の修正箇所や修正内容を示すタグが用いられている。例えば、青字で文字を記入したことを表す [f-blue]、赤字で文字を記入したことを表す [f-red]、取り消し線を付けたことを表す [sline] などのタグがある。本研究では取消線を表す [sline] タグ内の文字列を削除し、他のタグ (例: [f-blue] や [f-red]) はそのタグのみを除去して、タグのない修正文を作成する。Lang-8 コーパスにおける原文、修正文、タグを除去した修正文の例を表 3.2 に示す。

一般に、Lang-8 コーパスにおいて「原文」や「訂正文」として収録されているテキストは複数の文から構成されている。そこで、学習者が書いたテキストとネ

イティブスピーカーらが修正したテキストを句点によって分割し、原文 (誤りを含む文) と訂正文 (文法的に正しい文) を 1 対 1 に対応付ける。すなわち、文単位で誤りの訂正事例を収録する。また、文の対応付けの処理の後、原文と訂正文が一致している場合には、すなわち訂正が行われていない場合には、その文の組をデータセットから除去する。

上記の一連の手続きにより誤り文と訂正文の組の集合を得る。

3.3.2 誤りタイプのラベル付け

前項で構築されたデータセットにおける (誤り文, 訂正文) のペアに対し、誤りタイプをラベル付けする。付与する誤りタイプは 3.2 節で定義した 7 つの誤りタイプのいずれかとする。誤り文と訂正文を比較し、誤りタイプをルールベースの手法で推定する

誤りタイプのラベル付けを行うにあたり、前処理として、`diff_match_patch` ライブラリを用いて誤り文と訂正文の文字の対応付けを行い、両者が異なる箇所 (差異箇所) を特定する。誤り文における差異箇所は誤りが発生している箇所とみなすことができる。このライブラリは、文字列間の差異を効率的に検出することが可能である。特に、誤り文と訂正文の長さが異なる場合でも、差異箇所を正確に検出できる。本研究では、誤り文と訂正文の差異箇所が 2 箇所以上ある文の組をデータセットから除外する。これにより、データセットにおける誤り文には文法誤りが 1 つしか含まれないことが保証される。

次に、誤り文と訂正文を形態素解析ツール `MeCab` を用いて形態素解析する。これにより文は単語に分割され、また各単語の品詞の情報が得られる。また、前述の差異箇所検出結果と合わせると、誤りが発生した箇所の単語の品詞が得られる。この品詞の情報は誤りタイプを決定する際に利用される。

以下、7 つの誤りタイプのそれぞれについて、そのタイプを特定するルールベースの処理を説明する。

E1 助詞の特定　助詞の誤りを特定するために、本研究ではまず格助詞のリストを作成する。具体的には、日本語文法において重要な役割を果たす「は」「が」「に」「の」「で」「と」「へ」といった格助詞をリストに登録する。次に、誤り文と訂正文において以下の 3 種類のいずれかの差異が認められた場合、誤りタイプを「E1 助詞」と判定する。

1. 助詞の置換：誤り文中の助詞が訂正文で別の助詞に置き換えられた場合。
2. 助詞の追加：訂正文で新たに助詞が追加された場合。

3. 助詞の削除：誤り中の助詞が訂正文で削除された場合。

予備実験では、この方法による格助詞の誤りの検出の精度は非常に高いことが確認された。

上記の処理は格助詞の誤りを検出するための処理である。これに加え、接続助詞、終助詞、副助詞、間投助詞の誤りも検出する。誤り文と修正文の差異箇所における単語の品詞が「助詞」である場合、その誤り文のタイプも「E1 助詞」と判定する。

E2 助動詞の特定　検出した差異箇所における単語の品詞が「助動詞」のとき、誤りタイプを「E2 助動詞」とした。なお、助動詞の誤りには、助動詞が不足 (欠落) している場合と、助動詞が過剰に使用されている場合の 2 つのパターンが存在する。

E3 表記の特定　誤り訂正のデータセットにおける誤り文を調べたところ、表記誤りの多くがカタカナ表記とひらがな表記に関するものであった。この結果を踏まえ、表記誤りを検出するために、誤り文と訂正文の差異箇所を調べ、カタカナまたはひらがなの表記に差異が見られるかをチェックする。さらに、表記誤りを助詞の誤りや助動詞の誤りといった他の誤りタイプと明確に区別するため、カタカナまたはひらがなの表記の差異が名詞に出現するときのみ表記誤りと定義する。具体的には、以下の全ての条件を満たすとき誤りタイプを「E3 表記」と特定する。

1. 差異箇所が誤り文の形態素解析結果において「名詞」として認識されている。
2. 差異箇所が漢字を含まないカタカナ及びひらがなの表記である。
3. 誤り文と訂正文の編集距離が 1 である。これは 1 文字の誤りのみを表記誤りとし、2 文字以上の誤りは単語単位の誤りとみなして他の誤りタイプに分類するためである。

E4 動詞の活用誤りの特定　動詞の活用に誤りがあるかをチェックする。具体的には、以下の条件を全て満たすとき、誤りタイプを「E4 動詞の活用」と特定する。

1. 差異箇所が訂正文中の形態素解析結果において「動詞」として認識されている。
2. 差異箇所が動詞の語幹ではなく、語尾に該当している。
3. 差異箇所がひらがなである。

4. 誤り文と訂正文の編集距離が 1 である。(差異は 1 文字である)

5. 編集が文字の置換であるとき、すなわち誤り文の 1 文字が訂正文で別の 1 文字に置き換えられているとき、その 2 つの文字の母音が同じである (五十音表において同じ行の文字である) か子音が同じである (五十音表において同じ列の文字である)。

E5 動詞 (単語選択) の特定　誤り文と訂正文の差異箇所に該当する単語の品詞が「動詞」であり、かつ差異箇所が (動詞の活用語尾ではなく) 動詞の語幹に位置するとき、動詞の単語選択の誤りと判定する。

具体的には、以下の条件を全て満たすとき、誤りタイプを「E5 動詞 (単語選択)」と特定する。

1. 差異箇所の品詞が形態素解析において「動詞」として認識されている。

2. 差異箇所が動詞の語幹に該当する。

E6 名詞 (単語選択) の特定　誤り文と訂正文の差異箇所に該当する単語の品詞が「名詞」であり、かつ前述の E3(表記) の誤りタイプに該当しないとき、名詞の単語選択の誤りとみなす。既に述べたように、名詞 (単語選択) の誤りは、1 文字程度の書き誤りではなく、学習者が文脈や使用意図に適さない名詞を選択した場合を想定している。

具体的には、以下の条件を全て満たすとき、誤りタイプを「E6 名詞 (単語選択)」の誤りと特定する。

1. 差異箇所の品詞が「名詞」である。

2. 「E3 表記」の誤りの条件を満たさい。

E7 その他の特定　その他の誤りは、上記のいずれのルールにも該当しない場合に分類する。既に述べたように、この誤りタイプは、助詞、助動詞、表記、動詞活用、動詞 (単語選択)、名詞 (単語選択) といった明確に定義された誤りタイプに該当しないケースであり、分類が困難な多様な誤りを包括するカテゴリである。

上記で述べたルールベースの手法によって付与された誤りタイプの品質を調べるため、ランダムに抽出した 200 件のサンプルを用いて人手による評価を実施した。具体的には、それぞれの文対に付与された誤りタイプが正しいかどうかを人手で判定し、正解率 (誤りタイプが正しく分類できたサンプルの割合) を求めた。

その結果、正解率は 90.5%(181/200) となった。この結果は、ルールベースの手法が高い精度で誤りタイプを付与できることを示している。

表 3.3 は、構築したタイプ付き誤り訂正データベースにおいて、それぞれの誤りタイプとそれに該当するサンプル件数 (誤り文と訂正文の組の数) を示している。助詞の誤りが最も多いことがわかる。また、それに次いで動詞 (単語選択) 、動詞活用の誤りが多いことから、動詞もまた日本語学習者がよく犯す誤りであるといえる。全体でおよそ 23 万件のサンプルからなる大規模なデータセットが構築された。

表 3.3: タイプ付き誤り訂正データベースの統計

誤りタイプ	**件数**
E1 助詞	84,371
E2 助動詞	18,029
E3 表記	9,283
E4 動詞の活用	33,247
E5 動詞 (単語選択)	37,719
E6 名詞 (単語選択)	27,261
E7 その他	25,444
合計	**235,354**

3.4 誤りタイプの分類

本節では誤りタイプを分類する方法について述べる。3.4.1 項では、誤りタイプ分類モデルの学習について述べる。3.4.2 項では、モデル学習のための訓練データを増強するデータ拡張について述べる。

3.4.1 誤りタイプ分類モデルの学習

本研究では、事前学習済みの BERT および RoBERTa モデルのファインチューニングにより誤りタイプの分類モデルを学習する。具体的に使用した事前学習済みモデルは以下の通りである。これらは日本語のテキストから事前学習され、日本語文の意味理解に適応したモデルである。

- BERT: bert-base-japanese

- RoBERTa: rinna/japanese-roberta-base

7つの誤りタイプに分類するために、事前学習済みモデルBERTもしくはRoBERTaの最上位層に、7 クラスの出力層を持つ全結合層を追加する。ファインチューニングでは、誤り事例データベースを訓練データとして、BERT もしくはRoBERTaのパラメタ、および追加した全結合層のパラメタを更新する。具体的には、タイプ付き誤り訂正データベースにおける誤り文を入力、誤りタイプを出力として、損失関数が小さくなるようにパラメタを更新する。

誤りタイプ分類タスクの損失関数として交差エントロピー損失 (Cross-Entropy Loss) を使用する。この損失関数は以下の式で表される。

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{N} y_i \log \hat{y}_i \tag{3.3}$$

ここで、各記号の意味は以下の通りである。

- N はバッチサイズ。
- y_i はデータ i の正解ラベル (ワンホット形式で表現された確率分布)
- $\hat{y}_i$ はデータ i に対してモデルが予測するクラスの確率分布。

3.4.2 データ拡張

本研究で構築した誤り訂正事例データベースでは、表 3.3 に示したように、サンプル数が少ない誤りタイプがある。具体的には、「E3 表記」「E2 助動詞」「E6 名詞 (単語選択)」「E7 その他」のサンプル数が比較的少ない。サンプル数が少ない誤りタイプは分類が難しくなると考えられる。また、誤りタイプの数に大きな偏りがあると、学習した誤りタイプ分類モデルによる予測が多数の誤りタイプに偏り、正解率が低下する可能性がある。

そこで、「E7 その他」以外の誤りタイプのうち比較的数が少ない「E3 表記」「E6 名詞 (単語選択)」に着目し、これらの誤りの事例を人工的に生成するデータ拡張を行う。「E2 助動詞」のサンプル数が少ないにもかかわらずデータ拡張を行わなかった理由は、使役や受身の助動詞の使い分けなど助動詞の使用は文脈に大きく依存し、助動詞の誤用を機械的に生成すると、不自然な誤り文が生成される可能性があるためである。不自然なデータの追加は分類モデルの性能の低下を引き起こすリスクがある。

データ拡張は、文法的に正しい日本語文に対し、表記の誤りもしくは名詞選択の誤りを発生させることで行う。データ拡張の元となる日本語文のコーパスとしてKeiCO コーパスを用いる [11]。KeiCO コーパスは、日本語の敬語を含む文を収集し、敬語のレベル、書き言葉・話し言葉、尊敬語・謙譲語・丁寧語といった情報をアノテーションしたコーパスである。敬語は日本語学習者にとって修得するのが難しいことから、敬語を含む文は学習者の文法誤りを生じやすいと考え、KeiCO コーパスをデータ拡張の元テキストとして選択した。以下、表記誤りと名詞の単語選択の誤りを発生させる手法について述べる。

表記誤りのデータ拡張　正しい文に含まれるカタカナおよびひらがなを操作し、誤りを人工的に生成する。具体的には次の 2 つの操作を行う。

1. **ランダムな置換**：ひらがなやカタカナを母音が同じである別の仮名にランダムに置換する。ただし、助詞や動詞の活用語尾に該当する文字は、助詞の誤りや動詞の活用の誤りとの混同を避けるため、置換の対象としない。

2. **ランダムな削除**：ひらがなやカタカナのうち、助詞や動詞の活用語尾ではない文字をランダムに削除する。

置換操作と削除操作の適用確率をそれぞれ 70%、30%と設定し、この確率にしたがってどちらかの操作を選択する。次に、選択した操作にしたがって誤り文を生成する。生成した誤り文と元の文を新しい誤り訂正事例としてデータセットに追加する。

名詞 (単語選択) のデータ拡張　まず、文の形態素解析を行い、文中の名詞を抽出する。次に、抽出した名詞の中からランダムに 1 つの名詞を選択する。最後に、選択された名詞をその類義語に置き換える。類義語は日本語 WordNet によって得る。複数の類義語があるときはランダムに 1 つを選択する。これにより、文脈上適切でない単語が意図的に生成され、名詞選択の誤りを再現できる。先ほどと同様に、生成した誤り文と元の文を新しい誤り訂正事例としてデータセットに追加する。

3.5　誤り訂正モデルの学習

本節は誤り訂正モデルの学習について述べる。既に述べたように、誤り訂正モデルは、誤り文を入力、訂正文を出力とする系列変換モデルである。本研究では、系列変換に適応した事前学習済み言語モデルである T5 をファインチューニングすることで、誤り訂正モデルを得る。具体的には以下の T5 モデルを用いる。

- T5-large 日本語モデル: retrieva-jp/t5-large-long

このモデルは Retrieva 社が公開した日本語特化型の T5 モデルであり、大規模な日本語データを用いて事前学習されている。特に、長文入力への対応力が高く、誤り訂正のような文脈情報を重視するタスクに適している。

T5 モデルのファインチューニングにはタイプ付き誤り訂正データベースを用いる。同データセットの誤り文を入力、訂正文を出力として、ファインチューニングを行う。ただし、ファインチューニングは誤りタイプ毎に個別に行う。すなわち、誤り訂正事例データベースを誤りタイプによって 7 つに分割し、分割されたデータセットを用いて、誤りタイプに特化した誤り訂正モデルを 7 つ学習する。

第4章　評価

本章では提案手法の評価実験について述べる。4.1 節では誤りタイプ分類の性能を評価する。4.2 節では文法誤り訂正の性能を評価する。

4.1　誤りタイプ分類の評価

4.1.1　実験条件

タスクの定義

文法誤りのタイプを判定する 2 つのタスクを定義する。

誤り分類タスク　誤りを含む文が 7 種類の誤りタイプ (助詞、助動詞、表記、動詞活用、動詞 (単語選択)、名詞 (単語選択)、その他) のいずれに該当するかを判定する。7 値分類タスクである。入力される文は文法誤りを含んでいると仮定し、その誤りのタイプを分類する。

誤り検出・分類タスク　文が誤りを含むか否か、誤りを含む場合には 7 種類の誤りタイプのどれに該当するかを判定する。分類クラスとして「誤りなし」を追加し、8 値分類のタスクと定義する。このタスクでは文法的に正しい文、文法誤りを含む文の両方が入力される。

データセット

本実験では 3.3 節で構築した「タイプ付き誤り訂正データセット」を実験に用いる。同データセットは誤り文、訂正文、誤りタイプの組から構成される。誤りタイプ分類の実験では、このうち誤り文と誤りタイプの情報を用いる。

実験では、タイプ付き誤り訂正データセットを 80%、10%、10%に分割し、それぞれ訓練データ、開発データ、テストデータとして使用する。訓練データは誤りタイプ分類モデルの学習、すなわち BERT や RoBERTa のファインチューニングに用いる。開発データはエポック数の最適化に用いる。エポック毎に開発データで

のモデルの性能を測り、最も良い性能が得られたエポック数のモデルを選択する。テストデータは分類モデルの評価に用いる。

誤り検出・分類タスクでは、上記のデータセットに文法的に正しい文を追加する。具体的には、Lang-8 コーパスからネイティブスピーカーによって修正された文、すなわち文法的に正しい文を 100,000 件選択し、「誤りなし」のラベルを付与してデータセットに加えた。100,000 件の 80%を訓練データ、10%を開発データ、10%をテストデータに追加した。

誤りタイプ分類実験のデータセットの統計を表 4.1 に示す。

表 4.1: 誤りタイプ分類実験のデータセット

誤りタイプ	**訓練**	**開発**	**テスト**	**合計**
E1 助詞	67,497	8,437	8,437	84,371
E2 助動詞	14,423	1,803	1,803	18,029
E3 表記	7,426	928	929	9,283
E4 動詞活用	26,597	3,325	3,325	33,247
E5 動詞 (単語選択)	30,175	3,772	3,772	37,719
E6 名詞 (単語選択)	21,809	2,726	2,726	27,261
E7 その他	20,355	2,544	2,545	25,444
E8 誤りなし	80,000	10,000	10,000	100,000
合計	268,282	33,535	33,537	**335,354**

さらに、3.4.2 項で述べたように、本研究では「表記」及び「名詞 (単語選択)」の誤りタイプの分類性能を向上させるために、これら 2 つの誤りタイプの事例に対するデータ拡張を行った。拡張データの数を表 4.2 に示す。拡張された事例は全て訓練データに追加する。ただし、拡張データは誤り検出タスクのモデルの学習に使用し、誤り検出・分類タスクのモデルの学習には使用しない。実験時間が限られていたため、データ拡張後の訓練データを用いて「E8 誤りなし」を含めた誤りタイプを分類するモデルを学習することは省略した。誤り検出・分類タスクについてデータ拡張の効果を検証することは今後の課題である。

表 4.2: データ拡張によって追加された事例数

誤りタイプ	**拡張データ件数**
E3 表記	9,438
E6 名詞 (単語選択)	6,682

タイプ付き誤り訂正データセットでは、誤りタイプは自動的に付与されている。

前述のテストデータにおける誤りタイプも同様に自動付与されている。すなわち、正しい誤りタイプが必ず付与されているわけではない。誤り分類モデルの性能を正確に評価するために、少量のサンプルに対して人手で誤りタイプを付与した評価用データを作成する。テストデータからランダムに 150 件の誤り文を選択し、これに対して誤りタイプを人手で分類し、正解の誤りタイプを付与する。これを分類誤りタスクの評価に用いる。さらに、文法的に正しい文を 150 件用意し、「誤りなし」のラベルを付与して、データセットに追加する。これを誤り検出・分類タスクの評価に用いる。以下、上記の評価用データセットを「人手評価データ」と呼ぶ。人手評価データの統計を表 4.3 に示す。

表 4.3: 人手評価データの統計

誤りタイプ	**誤り分類タスク**	**誤り検出・分類タスク**
E1 助詞	45	45
E2 助動詞	10	10
E3 表記	13	13
E4 動詞活用	22	22
E5 動詞 (単語選択)	27	27
E6 名詞 (単語選択)	17	17
E7 その他	16	16
E8 誤りなし	—	150
合計	**150**	**300**

ファインチューニング時のハイパーパラメータ設定

本研究では、全ての誤りタイプ分類モデルにおいて、ファインチューニング時のハイパーパラメータ設定を統一する。具体的には、バッチサイズを 8、学習率 $1e^{-5}$ と設定する。 エポック数については開発データを用いて最適化する。最大エポック数を 5 に設定し、開発データに対するモデルの分類性能が最も高くなるエポック数を選択する。

評価指標

本研究では、誤りタイプ分類の結果を評価するために、以下の 3 つの指標を用いる。

精度 (precision) モデルがある誤りタイプに該当すると予測したデータのうち、正しいものの割合である。以下の式で定義される。

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.1}$$

ここで、TP は真陽性 (True Positive)、FP は偽陽性 (False Positive) の数を表す。

再現率 (recall) データセットにおけるある誤りタイプのデータのうち、モデルによってそのタイプに該当すると正しく分類できたものの割合である。以下の式で定義される。

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.2}$$

ここで、FN は偽陰性 (False Negative) の数を表す。

F1 スコア (F1-score) 精度 (Precision) と再現率 (Recall) の調和平均である。以下の式で定義される。

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

精度、再現率、F1 スコアは誤り分類のクラス (誤りタイプ) 毎に測る。これにより誤りタイプ分類モデルがそれぞれの誤りタイプを検出する能力を個別に評価する。さらに、誤り分類タスクでは 7 つのクラス、誤り検出・分類タスクでは 8 つのクラスの各指標のマクロ平均を計算し、誤りタイプ分類モデルの全体的な性能を評価する。

4.1.2 結果と考察

テストデータによる評価

テストデータに対する誤り分類タスクの精度、再現率、F1 スコアを表 4.5 に示す。この表は、分類モデルとして BERT および RoBERTa を用いたときの結果を載せている。また、データ拡張を行ったときと行わなかったときの結果も示している。

BERT と RoBERTa を比較すると、全体的に RoBERTa の方が優れている。F1 スコアのマクロ平均は、データ拡張をしないとき、BERT は 0.3747 であるのに対し、RoBERTa は 0.4817 であり、およそ 10 ポイント上回っている。データ拡張をしたときでも、RoBERTa は BERT と比べて F1 スコアがおよそ 4 ポイント高い。

表 4.4: テストデータに対する誤り分類タスクの結果

誤り	**データ**	**精度**		**再現率**		**F1 スコア**	
タイプ	**拡張**	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa
E1	なし	0.4938	0.6127	0.5002	0.7832	0.4970	0.6876
	あり	0.5942	0.6049	0.7169	0.7800	0.6498	0.6814
E2	なし	0.3934	0.4912	0.3827	0.4176	0.3880	0.4514
	あり	0.4695	0.5079	0.3716	0.3766	0.4149	0.4325
E3	なし	0.4115	0.5695	0.2128	0.3564	0.2805	0.4384
	あり	0.4524	0.5467	0.2936	0.3284	0.3561	0.4103
E4	なし	0.4050	0.4782	0.5239	0.4857	0.4569	0.4819
	あり	0.4296	0.4507	0.4394	0.4938	0.4344	0.4713
E5	なし	0.4357	0.5310	0.3712	0.3311	0.4009	0.4079
	あり	0.4107	0.4677	0.3706	0.3682	0.3896	0.4120
E6	なし	0.2428	0.4556	0.1960	0.4076	0.2169	0.4302
	あり	0.3639	0.5011	0.3062	0.2935	0.3326	0.3702
E7	なし	0.3304	0.4846	0.4542	0.4574	0.3825	0.4706
	あり	0.4449	0.4714	0.3970	0.4853	0.4196	0.4782
平均	**なし**	**0.3875**	**0.5174**	**0.3773**	**0.4627**	**0.3747**	**0.4817**
	あり	**0.4526**	**0.5072**	**0.4136**	**0.4465**	**0.4281**	**0.4651**

E1=助詞、E2=助動詞、E3=表記、E4=動詞の活用、E5=動詞 (単語選択)、E6=名詞 (単語選択)、E7=その他

また、誤りタイプ毎に F1 スコアを比較すると、どの誤りタイプでも RoBERTa は BERT を上回っている。特に誤りタイプが「E1 助詞」「E3 表記」「E6 名詞 (単語選択)」のときに差が大きい。精度については、いずれの誤りタイプでも RoBERTa の方が高い。再現率については、「E4 動詞の活用」「E5 動詞 (単語選択)」「E6 名詞 (単語選択)」で BERT の方が高いケースが見られるものの、マクロ平均では RoBERTa の方が明らかに高い。以上から、誤りのタイプ分類には BERT より RoBERTa の方が適していると言える。

次に、データ拡張の効果について考察する。BERT については、データ拡張によって F1 スコアのマクロ平均が向上したことから、データ拡張が効果的であった。データ拡張によって訓練事例を追加した 2 つの誤りクラスに着目すると、「E3 表記誤り」に対する F1 スコアが 0.2805 から 0.3561 へと向上し、「E6 名詞 (単語選択)」に対しても 0.2169 から 0.3326 へと大幅に改善した。この結果から、BERT

を誤り判定モデルとして使用したとき、データ拡張は有効であり、既存のデータセットにない新しい誤りのパターンを訓練データに加えることができたと考えられる。一方、RoBERTa についてはデータ拡張によって F1 スコアのマクロ平均が低下した。「E3 表記誤り」の F1 スコアは 0.4384 から 0.4103 へと低下し、「E6 名詞 (単語選択)」についても 0.4302 から 0.3702 へと悪化した。特に、再現率の低下が大きく、誤り検出の網羅性が損なわれていることが確認された。また、精度よりも再現率の低下が大きいことから、文法的に正しい文を誤り文と誤検出するエラーよりも、誤り文を検出できなかったエラーの方が増えたことがわかった。したがって、RoBERTa においてはデータ拡張は効果的ではないと言える。RoBERTa はデータ拡張をしないデータセットによるファインチューニングでも誤りタイプの分類に必要な知識を十分に学習できており、データ拡張によって正しくない訓練事例が追加されたことによって誤りタイプ分類の性能が低下した可能性がある。以上から、BERT と RoBERTa においてデータ拡張の効果が異なることが明らかになった。

テストデータに対する誤り検出・分類タスクの精度、再現率、F1 スコアを表 4.5 に示す。分類クラスに「E8 誤りなし」が追加されており、モデルが入力文における文法誤りの有無を判定する能力も評価されている。また、4.1.1 項で述べたように、データ拡張したデータセットを用いて学習したモデルは評価していない。

表 4.5: テストデータに対する誤り検出・分類タスクの結果

誤り	**精度**		**再現率**		**F1 スコア**	
タイプ	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa
E1	0.6057	0.5941	0.6223	0.5520	0.6139	0.5723
E2	0.5126	0.5211	0.3548	0.2485	0.4194	0.3365
E3	0.5245	0.5262	0.3523	0.2794	0.4215	0.3650
E4	0.5131	0.4738	0.4270	0.3541	0.4661	0.4053
E5	0.4718	0.4520	0.3543	0.2040	0.4047	0.2811
E6	0.4051	0.3946	0.2649	0.1807	0.3203	0.2479
E7	0.4808	0.4945	0.3371	0.2678	0.3964	0.3474
E8	0.5994	0.5488	0.7543	0.8274	0.6680	0.6599
平均	**0.5141**	**0.5006**	**0.4334**	**0.3642**	**0.4638**	**0.4019**

E1=助詞、E2=助動詞、E3=表記、E4=動詞の活用、E5=動詞 (単語選択)、E6=名詞 (単語選択)、E7=その他、E8=誤りなし

まず、BERT と RoBERTa を比較する。BERT の平均精度、再現率、F1 スコア

はそれぞれ0.5141、0.4334、0.4638であるのに対し、RoBERTaは0.5006、0.3642、0.4019であり、3つの指標のいずれもBERTはRoBERTaを上回る結果が得られた。また、「E8 誤りなし」の分類に着目すると、BERTのF1スコアは0.6680であり、RoBERTaの0.6599をわずかに上回っていることから、文法誤りの有無の判定についてもBERTの方が優れていることがわかった。誤りタイプごとに評価指標を比較すると、BERTはどの誤りタイプに対しても精度や再現率に大きな差はない。一方、RoBERTaは「E5 動詞 (単語選択)」「E6 名詞 (単語選択)」に対する再現率が低く、それに伴いF1スコアも他の誤りタイプに比べて大きく低下しており、誤りタイプに対する分類性能のばらつきが見られる。以上の結果から、誤り検出タスクとは異なり、誤り検出・分類タスクではRoBETAaよりBERTの方が適していると言える。誤りタイプを分類するだけではなく文法誤りの有無の判定を含むという条件では、BERTはどの誤りタイプに対しても安定した性能を発揮している。

4.1.3 人手評価データによる評価

人手評価データに対する誤り分類タスクの精度、再現率、F1スコアを表4.6に示す。人手評価データは人手で正解の誤りタイプを付与しているため、モデルの正確な比較ができる。一方、データ数は150件とこれまで述べたテストデータと比べて小さいため、各指標を有効数字2桁で示している。

BERTとRoBERTaを比較すると、テストデータでの実験結果 (表4.4) と同じように、全体的にRoBERTaの方が優れていることが確認された。データ拡張をしない条件のときの「E4 動詞の活用」に対する再現率ならびにF1スコアを除いて、どの誤りタイプでもRoBERTaの精度・再現率・F1スコアはBERTを上回り、F1スコアのマクロ平均でもRoBERTaはBERTより0.12ポイント (データ拡張なしのとき) または0.14ポイント (データ拡張ありのとき) 高かった。

次にデータ拡張の効果について考察する。F1スコアのマクロ平均を見ると、BERTではデータ拡張によって指標が改善しなかったが、RoBERTaではわずかに (2ポイント) 高かった。データ拡張によって訓練事例を追加した「E3 表記」について、BERTモデルではF1スコアが0.24から0.27にわずかに向上し、RoBERTaでは0.32から0.47へと大幅な改善が確認された。再現率においても、BERTは0.15から0.23に、RoBERTaは0.23から0.30に向上しており、表記誤りに対するデータ拡張の効果が顕著に現れている。この結果は、表記誤りに特化したデータが、モデルの表記誤りの検出能力を向上させたことを示している。一方、「E6 名詞 (単語選択)」では、両モデルで性能が低下した。BERTはF1スコアが0.40から0.32に、RoBERTaは0.48から0.42に低下しており、データ拡張が名詞の単語選択の

表 4.6: 人手評価データに対する誤り分類タスクの結果

誤り	データ	精度		再現率		F1 スコア	
タイプ	拡張	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa
E1	なし	0.57	0.66	0.87	0.89	0.69	0.75
	あり	0.50	0.64	0.82	0.93	0.62	0.76
E2	なし	0.50	1.00	0.20	0.70	0.29	0.82
	あり	0.67	0.67	0.40	0.60	0.50	0.63
E3	なし	0.50	0.50	0.15	0.23	0.24	0.32
	あり	0.33	1.00	0.23	0.30	0.27	0.47
E4	なし	0.52	0.53	0.55	0.45	0.53	0.49
	あり	0.50	0.62	0.50	0.59	0.50	0.60
E5	なし	0.50	0.65	0.48	0.63	0.49	0.64
	あり	0.55	0.67	0.41	0.67	0.47	0.67
E6	なし	0.46	0.58	0.35	0.41	0.40	0.48
	あり	0.50	0.71	0.24	0.29	0.32	0.42
E7	なし	0.67	0.53	0.50	0.63	0.57	0.57
	あり	0.64	0.63	0.44	0.63	0.52	0.63
平均	**なし**	**0.54**	**0.63**	**0.44**	**0.56**	**0.46**	**0.58**
	あり	**0.53**	**0.70**	**0.44**	**0.57**	**0.46**	**0.60**

E1=助詞、E2=助動詞、E3=表記、E4=動詞の活用、E5=動詞 (単語選択)、E6=名詞 (単語選択)、E7=その他

誤りの分類性能に悪影響を及ぼしている。この結果は、拡張データの品質やその適用範囲が、この誤りの特性に十分対応していなかったことを示している。さらに、データ拡張によって訓練事例を追加していない誤りタイプについても精度や再現率に変化が見られた。例えば、「E1 助詞」について、BERT では F1 スコア 0.69 から 0.62 に低下した一方で、RoBERTa では 0.75 から 0.76 に改善した。また、「E2 助動詞」については、BERT については F1 スコアが 0.29 から 0.50 に向上したが、RoBERTa では 0.82 から 0.63 に低下した。データ拡張によって直接的に訓練事例が追加されない誤りタイプについても、その誤り分類の性能が間接的に影響を受けていることがわかる。とはいえ、人手評価データによる評価では、モデルや誤りタイプによっては F1 スコアの低下を招いたものの、全体的に見てデータ拡張は効果的であったと言える。

人手評価データに対する誤り検出・分類タスクの精度、再現率、F1 スコアを表

4.7 に示す。分類クラスに「E8 誤りなし」が追加されている。

表 4.7: 人手評価データに対する誤り検出・分類タスクの結果

誤り	**精度**		**再現率**		**F1 スコア**	
タイプ	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa
E1	0.63	0.55	0.73	0.62	0.68	0.58
E2	0.71	1.00	0.50	0.60	0.59	0.75
E3	0.40	0.40	0.15	0.15	0.22	0.22
E4	0.65	0.40	0.50	0.45	0.56	0.51
E5	0.83	0.80	0.56	0.44	0.67	0.57
E6	0.43	0.17	0.18	0.06	0.25	0.09
E7	0.43	0.55	0.38	0.38	0.40	0.44
E8	0.73	0.68	0.88	0.86	0.80	0.76
平均	**0.61**	**0.60**	**0.49**	**0.43**	**0.55**	**0.49**

E1=助詞、E2=助動詞、E3=表記、E4=動詞の活用、E5=動詞 (単語選択)、E6=名詞 (単語選択)、E7=その他、E8=誤りなし

BERT の平均 F1 スコアは 0.55 であり、RoBERTa の 0.49 を上回っている。特に、「E1 助詞」「E5 動詞 (単語選択)」「E6 名詞 (単語選択)」といった誤りタイプで両者の差が大きい。一方で、「E5 動詞 (単語選択)」や「E3 表記」では両モデルとも F1 スコアが低かった。特に「E3 表記」では BERT と RoBERTa の平均 F1 スコアは共に 0.22 と低い値にとどまった。この結果は、これらの誤りタイプの検出および分類が依然として難しい課題であることを示している。また、RoBERTa は「E6 名詞 (単語選択)」に対する F1 スコアのマクロ平均が 0.09 とかなり低く、この誤りタイプの特徴をうまく学習できていないことが伺える。文法誤りの有無を検出する性能は「E8 誤りなし」に対する精度、再現率、F1 スコアで評価できるが、BERT、RoBERTa ともに再現率や F1 スコアが十分に高いことが確認できた。また、全ての指標で BERT が RoBERTa を上回っていることから、誤り検出の能力でも BERT の優位性が確認された。以上をまとめると、誤り検出・分類タスクにおいて、BERT は RoBERTa より適していると結論付けられる。これはテストデータでの実験結果 (表 4.5) から得られる結論と一致している。

4.2 誤り訂正の評価

4.2.1 実験条件

データセット

誤り訂正の実験でも、3.3 節で構築したタイプ付き誤り訂正データベースを使用する。誤りタイプ分類の実験と同様に、各誤りタイプごとに、全データのうち80%を訓練データ、10%を開発データ、10%をテストデータとして使用する。データセットの統計は表 4.1 と同じである。訓練データは誤り訂正モデルの学習、すなわち事前学習済み T5 モデルのファインチューニングに用いる。誤りタイプ毎にデータセットを分割し、誤りタイプ毎に誤り訂正モデルを学習する。また、提案手法との比較のため、全ての訓練データを用いて、誤りタイプを区別せずに誤りを訂正するモデルも学習する。開発データはエポック数の最適化のために用いる。テストデータは誤り訂正モデルの評価に用いる。タイプ付き誤り事例データベースにおける訂正文は Lang-8 コーパスから収集したものであり、誤り文を正しく訂正した文であるため、誤りタイプ分類の実験のように人手による評価データは作成せず、このテストデータだけを評価に用いる。また、入力される文は文法誤りを必ず含むものとし、文法的に正しい文は入力されないものと仮定する。

比較手法

本研究では、文法誤り訂正の性能を評価するために、以下の手法を比較した。

ベースライン BL ベースラインモデルは、誤りタイプを区別せずに誤り訂正を行うモデルである。先ほど述べたように、全ての訓練データを用いて単一の誤り訂正モデルを学習する。機械学習により文法誤り訂正を実現する最も基本的な手法である。

提案手法システム $\mathbf{PRO_{gold}}$ まず、あらかじめ誤りタイプ毎に誤り訂正モデルを学習する。次に、誤り文が入力されたとき、その誤りタイプを決定し、その誤りタイプの誤り訂正モデルを用いて訂正文 (誤りを修正した文) を出力する。ただし、本システムは誤りタイプを推定せず、タイプ付き誤り訂正データベースに付与された誤りタイプを用いる。すなわち、誤りタイプの分類が正しく行われたという仮定の下で提案システムの性能を評価する。

提案手法システム $\mathbf{PRO_{auto}}$ $\mathrm{PRO_{gold}}$ と同じく提案手法による誤り訂正システムである。ただし、誤りタイプは、3.4 節で提案した誤りタイプ分類モデルを

用いて決定する。具体的には、誤りタイプ分類モデルとして、データ拡張後の訓練コーパスを用いて学習した RoBERTa モデルを使用した。4.1 節で報告したように、このモデルが誤り分類タスクの人手評価データに対する F1 スコアが最も高かったためである。

ファインチューニング時のハイパーパラメータ設定

本研究では、全ての文法誤り訂正モデルのファインチューニングにおいて同一のハイパーパラメータを使用する。具体的には、バッチサイズを 8、学習率を $5e^{-5}$ に設定する。エポック数については開発データを用いて最適化する。最大エポック数を 5 に設定し、開発データに対するモデルの誤り訂正の性能が最も高くなるエポック数を選択する。

評価指標

本研究では、文法誤り訂正タスクの評価指標として **GLEU**(Grammar-aware Language Evaluation Understudy) を採用する [14]。GLEU は、機械翻訳の評価指標である BLEU を基に設計されており、文法誤り訂正の性能を適切に評価できるように特化されている。

GLEU は、生成された訂正文と正解文 (リファレンス) の間で一致する単語 n-gram の割合を基にスコアを算出する指標である。訂正文がどの程度リファレンスに近いかを評価すると同時に、誤り文と過剰に一致していないかを評価する。これにより、文法誤り訂正タスクにおいて重要な「正確性」と「修正の適切性」の両方をバランスよく評価できる。

GLEU スコアは以下の式で定義される。

$$\mathrm{GLEU}(C, R, S) = BP \cdot \exp\left(\sum_{n=1}^{4} w_n \log p'_n\right)$$

ここで、各記号の意味は以下の通りである：

- C：生成された訂正文。
- R：リファレンス文 (正しい文)。
- S：誤りを含む元の入力文。
- p'_n：修正された n-gram 精度。訂正文とリファレンス文の n-gram の重複度を測る。

- w_n：各 n-gram の重み (通常は $\frac{1}{4}$)。
- BP：Brevity Penalty。生成された訂正文の長さが短すぎるときに与えるペナルティ。

p'_n は次のように定義される。

$$p'_n = \frac{\sum_{n\text{-}gram \in C} Count_{R\setminus S}(n\text{-}gram) - \lambda(Count_{S\setminus R}(n\text{-}gram)) + Count_R(n\text{-}gram)}{\sum_{n\text{-}gram' \in C'} Count_S(n\text{-}gram') + \sum_{n\text{-}gram \in R\setminus S} Count_{R\setminus S}(n\text{-}gram)}$$

- $Count_{R\setminus S}(n\text{-}gram)$：リファレンス文に存在し、かつ元の誤り文に含まれない n-gram の出現回数。
- $Count_{S\setminus R}(n\text{-}gram)$：元の誤り文に存在し、かつリファレンス文に含まれない n-gram の出現回数。
- $Count_R(n\text{-}gram)$：リファレンス文に含まれる n-gram の出現回数。
- $Count_S(n\text{-}gram')$：元の誤り文に含まれる n-gram の出現回数。
- $Count_{R\setminus S}(n\text{-}gram)$：リファレンス文に存在し、かつ元の誤り文には含まれない n-gram の出現回数。
- λ：訂正されなかった誤りの n-gram に対するペナルティの強さを調整する係数 (通常は 1)。

4.2.2 結果と考察

表 4.8 は、BL(ベースライン)、PRO_{gold}(正解の誤りタイプを用いた提案システム)、PRO_{auto}(自動推定された誤りタイプを用いた提案システム) の誤りタイプ毎の GLEU スコアを示している。

結果を見ると、PRO_{gold} の GLEU スコアは「E1 助詞」以外の誤りタイプにおいて BL を上回っており、全体のスコアもベースラインの 0.7618 に対して 0.7739 と高い値を示している。これは、それぞれの誤りタイプに特化した誤り訂正モデルを学習し、入力の誤りタイプに応じてこれを適切に使い分けることによって、文法誤り訂正の性能が向上することを示している。ただし、誤りタイプが「E1 助詞」のとき、PRO_{gold}(0.7884) が BL(0.7912) をわずかに下回っており、ベースラインモ

表 4.8: 誤り訂正の実験結果

誤りタイプ	BL	PRO_{gold}	PRO_{auto}
E1 助詞	0.7912	0.7884	0.7763
E2 助動詞	0.8076	0.8327	0.7627
E3 表記	0.7442	0.7892	0.7354
E4 動詞の活用	0.7481	0.7501	0.7107
E5 動詞 (単語選択)	0.7066	0.7270	0.6763
E6 名詞 (単語選択)	0.7504	0.7704	0.7260
E7 その他	0.7399	0.7699	0.7242
全て	**0.7618**	**0.7739**	**0.7390**

デルの優位性を示している。一方、「E2 助動詞」や「E3 表記」の誤りタイプについては、PRO_{gold} は BL を顕著に上回っている。これらの誤りタイプはデータセットにおいてデータ数が少なく、全体で 1 つの誤り訂正モデルを学習するベースラインでは訂正が難しかったが、誤りタイプ毎に特化した誤り訂正モデルを用いることでこれらの誤りに対する訂正能力が向上した。

一方、PRO_{auto} の全体の GLEU スコアは 0.7390 となり BL よりやや低かった。特に誤りタイプが「E2 助動詞」のときは、PRO_{auto} の GLEU スコアは BL と比べて大きく低下した。これは誤りタイプ分類モデルの性能が低いことが原因と考えられる。今回の実験に使用した誤りタイプ分類モデルの F1 スコアは、テストデータに対して 0.4651、人手評価データに対して 0.60 程度であり、十分に高いとは言えない。誤りタイプの分類に失敗すると、正しいタイプに対応した誤り訂正モデルが適用できず、このことが GLEU スコアの低下を招いたと推測できる。以上をまとめると、提案手法は誤りタイプの分類が正しいという条件下では誤り訂正の性能が向上したが、誤りタイプを自動推定するという条件下では性能は向上せず、誤りタイプ分類モデルの性能向上が課題として残った。

第5章　おわりに

5.1　本研究のまとめ

本研究では、日本語学習者が作成した文章に含まれる文法的誤りを自動的に訂正するための手法を提案し、その効果を評価した。従来の誤り訂正モデルでは、タイプ毎の誤りの特性が十分に考慮されていないという課題があった。本研究では、この課題に対応するため、誤りのタイプ分けと個々の誤りタイプに特化した誤り訂正モデルを組み合わせた新たなアプローチを提案した。

具体的には、まず先行研究を参考に、「助詞」「助動詞」「表記」など 7 種類の誤りタイプを定義した。次に、誤りを含む文が入力されたとき、その誤りのタイプを推定した。最後に、それぞれの誤りタイプ毎に誤り訂正モデルを学習し、推定された誤りタイプに対応する誤り訂正モデルを用いて学習者の文法誤りを訂正した。最終的に誤りを訂正した文と誤りタイプの両方を学習者に提示するシステムを考案した。

上記の提案システムを以下の手順で実装した。まず、Lang-8 コーパスから誤り文と訂正文の組を収集し、ルールベースの手法によって個々の組に対して誤りタイプを付与することで、タイプ付き誤り訂正データセットを構築した。次に、タイプ付き誤り訂正データセットを用いて事前学習済み BERT または RoBERTa をファインチューニングすることで、誤りタイプを分類するモデルを学習した。さらに、データセット内でサンプル数の少ない「表記」と「名詞 (単語選択)」の誤りタイプについて、誤りの事例を合成し、データセットに加えるデータ拡張を実施した。最後に、タイプ付き誤り訂正データセットを誤りタイプ毎に分割し、誤りタイプ毎に事前学習済み T5 モデルをファインチューニングすることで、個々の誤りタイプの特性を十分に考慮して誤り文を正しい文に変換する誤り訂正モデルを学習した。

実験では、まず誤りタイプ分類モデルを評価した。分類モデルとして BERT もしくは RoBERTa を用いたとき、データ拡張を行うときと行わないときでモデルを比較した。誤り分類タスク (誤り文に対してその誤りタイプを分類するタスク) について、人手で正解の誤りタイプを付与した「人手評価データ」を用いた評価では、ほとんどの誤りタイプについて RoBERTa の F1 スコアは BERT よりも高

く、平均F1スコアでもRoBERTaはBERTを上回る結果が得られた。また、データ拡張によってRoBERTaの平均F1スコアが0.02ポイント改善した。誤り検出・分類タスク(誤りタイプに「誤りなし」を追加し、誤りを含むか否かの判定と誤りを含むときにそのタイプを分類するタスク)の評価では、RoBERTaの平均F1スコアはBERTよりも低く、タスクによって事前学習済みモデルの優劣に違いが見られた。

次に、提案手法による文法誤り訂正の性能を評価した。ベースラインモデルとして誤りタイプを事前に判定せず全ての誤りタイプの文を修正する単一の誤り訂正モデルを学習し、提案手法と比較した。文法誤り訂正の標準的な評価指標であるGLEUスコアを用いて評価したところ、正解の誤りタイプを用いたときの提案手法のGLEUスコアはベースラインよりも高いことを確認した。しかし、誤りタイプを自動分類したときの提案手法はベースラインと比べてGLEUスコアの改善は見られなかった。これは誤りタイプ分類モデルの性能が十分に高くないことが原因と考えられた。

本研究の意義としては以下の点が挙げられる。第一に、誤りタイプ分類モデルと誤りタイプに特化した誤り訂正モデルの組み合わせにより、誤りタイプに特化した柔軟な訂正を可能にした。第二に、サンプル数が少ない誤りタイプに対するデータ拡張の効果を実験的に検証し、実験条件によってはデータ拡張が誤りタイプ分類の性能向上に寄与することを示した。

5.2 今後の課題

本研究では、誤りのタイプを考慮して文法誤りを訂正する新しい手法を提案し、その有効性を示した。しかし、いくつか課題も残されている。今後の研究で解決すべき課題を以下に挙げる。

まず、誤りタイプ分類モデルの性能向上が挙げられる。本研究では、BERTやRoBERTaモデルを用いて誤りタイプを自動分類するアプローチを採用したが、特に「名詞(単語選択)」や「動詞(単語選択)」といった誤りの要因が複雑な誤りタイプにおいて、その分類精度が十分ではなかった。より精度の高い分類モデルの設計や、有効的な特徴量の追加を検討する必要がある。

次に、データ拡張手法の最適化が挙げられる。本研究では、「表記」や「名詞(単語選択)」のような訓練事例数が不足している誤りタイプに対してデータ拡張を適用した。しかし、一部の誤りタイプのみのデータ拡張ではその効果は限定的であった。拡張データの生成手法を改良し、その品質を改善することが求められる。特に、学習者の実際の誤り傾向を反映したより自然な拡張データの作成が重要である。

また、提案手法の評価にも課題が残されている。実験では、主にLang-8コーパスを基にしたデータセットで評価実験を行ったが、他のデータセットを用いた評価や、実際の学習環境下での評価も行うべきである。このような包括的な評価実験を通じて、提案手法の誤り訂正能力の汎用性を明らかにする必要がある。

最後に、誤り訂正モデルの解釈性とフィードバックの質の向上が挙げられる。本研究では、誤りの分類と訂正結果の提示に重点を置いたが、学習者がより直感的に理解しやすいフィードバック形式や、訂正の根拠を示す機能の開発が求められる。これにより、学習者が自身の日本語文法に関する知識を深めることに寄与するシステムの実現が期待される。

参考文献

[1] 新井美桜, 金子正弘, 小町守. 日本語学習者向けの文法誤り検出機能付き作文用例検索システム. 人工知能学会論文誌, pp. A–K23_1–9, 2020.

[2] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 793–805, 2017.

[3] Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. A neural grammatical error correction system built on better pre-training and sequential transfer learning. In *Proceedings of the ACL 2019 Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 213–227, 2019.

[4] Daniel Dahlmeier and Hwee Tou Ng. Grammatical error correction with alternating structure optimization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 915–923, Portland, Oregon, USA, 2011. Association for Computational Linguistics.

[5] R. Dale and A. Kilgarriff. Helping our own: The hoo 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pp. 242–249, 2011.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[7] 甫立健悟, 金子正弘, 勝又智, 小町守. 文法誤り訂正における訂正度を考慮した多様な訂正文の生成. 自然言語処理, 第 28 巻, 第 2 号, pp. 428–449, 2021.

[8] Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4248–4254, Online, 2020. Association for Computational Linguistics.

[9] 小山碧海, 喜友名朝視顕, 小林賢治, 新井美桜, 三田雅人, 岡照晃, 小町守. 日本語文法誤り訂正のための誤用タグ付き評価コーパスの構築. 自然言語処理, Vol. 30, No. 2, pp. 330–371, 2023.

[10] Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Stella Tong. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3291–3301, Minneapolis, Minnesota, USA → Online, 2019. Association for Computational Linguistics.

[11] M. Liu, 小林一郎. 選択体系機能言語学に基づく日本語敬語コーパスの構築と検証. 言語処理学会第 28 回年次大会, pp. E7–4, 2022.

[12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[13] 水本智也, 小町守, 永田昌明, 松本裕治. 日本語学習者の作文自動誤り訂正のための語学学習 sns の添削ログからの知識獲得. 人工知能学会論文誌, pp. 420–432, 2013.

[14] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 588–593, Beijing, China, 2015. Association for Computational Linguistics.

[15] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits

of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[16] Guy Rotman, Omri Abend, and Amir Globerson. Efficient grammatical error correction via multi-task training and optimized training schedule. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6362–6368, Online, 2020. Association for Computational Linguistics.

[17] A. Rozovskaya and D. Roth. Building a state-of-the-art grammatical error correction system. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 2, pp. 231–235, 2014.

[18] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 2014.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017)*, pp. 5998–6008, Long Beach, CA, USA → Online, 2017. Curran Associates, Inc.

[20] Ziang Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 380–386, San Diego, California, 2016. Association for Computational Linguistics.

[21] Weimin Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jing Liu. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 156–165, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.