

学修番号 18860621

修士論文

少量のラベルデータを利用した文法誤り訂正

勝又 智

2020年2月21日

首都大学東京大学院
システムデザイン研究科 情報科学域

勝又 智

審査委員：

小町 守 准教授 (主指導教員)

山口 亨 教授 (副指導教員)

高間 康史 教授 (副指導教員)

少量のラベルデータを利用した文法誤り訂正*

勝又 智

修論要旨

近年、自然言語処理の分野において、言語学習者支援タスクとして、文法誤り訂正に関する研究が盛んに行われている。文法誤り訂正は、言語学習者の文法的に誤った文を入力とし、訂正された文を出力とする翻訳問題として取り組まれていることが多い。この問題設定の類似性から、機械翻訳で研究されている手法を文法誤り訂正に適用した研究が数多く存在している。また、機械翻訳の研究と同様に、文法誤り訂正では大量の対訳コーパスや擬似データを使用することで性能が改善されることが知られている。

文法誤り訂正において一般に使用することが可能なデータとしては、言語学習者の相互添削サイトである Lang-8 から抽出したものが知られている。このデータは、学習者の文とそれを訂正した文として対訳関係になっており、対訳コーパスとして使用可能である。Lang-8 から抽出したうち、もっともデータ量の多い言語は英語であるが、機械翻訳のデータ量が豊富な言語対と比べるとまだまだ少なく、英語以外の言語についてはさらに訓練データが少ない。そのため、この訓練データが少量であることが文法誤り訂正の問題の一つとなっている。実際、学習者の文は Lang-8 や HiNative などの言語学習者支援サイトから抽出することで、大量に獲得できるが、この誤り文に対応する訂正文を得ることは簡単ではない。Lang-8 に投稿されている訂正文は、本当に正しい訂正文であるのかは保証されておらず、正しい訂正文であることを保証するために、学習者の対象言語の母語話者が精査した場合、そのコストが莫大となってしまう。このように、誤り文と訂正文が対応した大規模対訳コーパスを作成することは困難である。

Bryant らは、この使用可能な学習者データが少量である問題に対して、使用可能な訓練データを制限した英語の文法誤り訂正のワークショップを開催した。この

*首都大学東京大学院 システムデザイン研究科 情報科学域 修士論文, 学修番号 18860621, 2020年2月21日.

ワークショップでは、大規模単言語コーパスに擬似誤りを付与して擬似対訳データを作成する手法などが提案された。

本研究では、主に機械翻訳で研究されている教師なし手法を文法誤り訂正に取り入れることで、使用可能な学習者データが少ない状況での実験を行った。具体的には、統計的機械翻訳を中心とした教師なし手法とニューラル機械翻訳を中心とした教師なし手法をそれぞれ用いて、文法誤り訂正モデルを構築した。

統計的機械翻訳を用いた手法は、誤り側と訂正側のフレーズ分散表現を共有したものを使用して、誤りフレーズとそれを訂正したフレーズの対応をとる。その後、このフレーズテーブルを2種類の更新方法を用いて修正していく。

ニューラル機械翻訳を用いた手法は、Transformer と呼ばれる自己回帰型注意機構を中心とした Encoder-Decoder ネットワークを使用する。この教師なし手法は、事前に誤り側と訂正側で言語知識を共有するように Masked Language Model と呼ばれる、人手のラベルを必要としない学習を行う手法である。その後、誤り側、訂正側について、Denoising Auto-Encoder と Back-Translation の目的関数を最小化するように同時に学習していく。Denoising Auto-Encoder はある文に対して表層的なノイズを加えて、それを Encoder-Decoder で修正する学習方法である。これにより、Transformer は誤り側、訂正側それぞれに対する表現能力が獲得されることが考えられる。Back-Translation は誤り側の文、または訂正側の文に対して現在学習中の誤り訂正モデルまたは誤り生成モデルで推論して、出力された訂正文、または誤り文を獲得する。この文対を用いて逆方向のモデルである、誤り生成モデルまたは誤り訂正モデルを訓練する学習方法である。この Back-Translation により、誤り文を訂正文に翻訳する能力が得られると期待される。

また、教師なし機械翻訳の手法は、対訳コーパスではなく、誤り側の文集合と訂正側の文集合となる、コンパラブルコーパスが訓練データとして必要になる。そのため、本研究では誤り文として機械翻訳の出力文を使用することで、擬似コンパラブルコーパスを作成した。具体的には、文法誤り訂正が対象としている言語の単言語コーパスを訂正側として、別の言語で書かれた単言語コーパスを対象言語に翻訳したデータを誤り側とした。

本研究では、作成した擬似コンパラブルコーパスと少量の学習者データを使用して文法誤り訂正モデルを構築し、その性能を評価する。評価データには英語学習者

支援タスクとして CoNLL-2014 test data, JFLEG test data, W&I+LOCNESS, 中国語学習者支援タスクとして NLPCC-2018 test data を使用する。統計的機械翻訳とニューラル機械翻訳の違いで、最終的な結果にどのような差が生じるのか考察した。また、統計的機械翻訳を用いた手法については、フレーズテーブルの更新手法による出力の差異について分析を行った。

本研究の貢献は以下の通りである。

- 文法誤り訂正で利用できる訓練データが少量である問題に対して、機械翻訳で研究されている教師なし手法を導入、その性能を検証した。
- 教師なし手法を導入する際に必要となるコンパラブルコーパスに、機械翻訳の出力を誤り側としたデータを使用し、その効果を検証した。

本論文の構成は以下のようになっている。第 1 章では本研究の提案、貢献、概要について述べる。第 2 章では、関連研究として、少量データに対する文法誤り訂正の研究や、機械翻訳で研究されている教師なし手法について述べる。第 3 章では、本研究で使用した擬似コンパラブルコーパスの構築や、教師なし文法誤り訂正について述べる。第 4 章では本研究の実験設定と実験結果を述べる。第 5 章では実験結果に対する分析、考察を行う。第 6 章では本研究のまとめについて述べる。

(Almost) Unsupervised Grammatical Error Correction*

Satoru Katsumata

Abstract

Research on grammatical error correction (GEC) has gained much attention recently. Many studies treat GEC as a task that involves translation from a grammatically erroneous sentence (source side) into a correct sentence (target side) and thus, leverage methods based on machine translation (MT) for GEC. For instance, some GEC systems use large parallel corpora and synthetic data.

One such large-scale parallel corpus is the public parallel corpus extracted from Lang-8. This corpus consists of sentences written by language learners on the source side and sentences corrected by native speakers on the target side. English learner sentences are the most common data in the data extracted from Lang-8. However, these English sentences are very small compared to a parallel corpus that has a large amount of data used in MT. The training data in other languages is even less. Hence, the lack of training data is one of the problems of GEC. In fact, it is easy to obtain the learner sentences by extracting them from the language learning social media platforms, such as Lang-8 and HiNative. On the other hand, it is not easy to obtain corrected sentences corresponding to these learner sentences. The sentence on the target side in Lang-8 is not necessarily completely corrected because annotators are learners of other languages; hence, they are not always experts in teaching. It is costly to have experts correct erroneous sentences. Therefore, it is difficult to prepare large parallel corpora in GEC.

*Master's Thesis, Department of Computer Science, Graduate School of Systems Design, Tokyo Metropolitan University, Student ID 18860621, February 21, 2020.

Bryant et al. held a GEC shared task, which limited the amount of annotated data available to participants and aimed to encourage the development of systems without large parallel corpora. Most systems in this shared task proposed the method of creating a synthetic parallel corpus by introducing pseudo errors to a large monolingual corpus.

We introduce an unsupervised method based on MT for GEC to address the issue of lack of a learner corpus. In particular, we use unsupervised statistical MT (SMT) and neural MT (NMT) to make the GEC model.

The SMT methods are based on phrase-based SMT and two phrase table refinements, i.e., forward and backward refinement. Forward refinement simply augments a learner corpus with automatic corrections whereas backward refinement expands both a learner corpus and a raw corpus to train GEC model using back-translation.

The NMT method is based on an Encoder–Decoder model using self-attention, called Transformer. This unsupervised method trains Masked Language Model (MLM) to obtain a deep representation towards the source and target side. Then, this method uses two objectives; Denoising Auto–Encoder (DAE) and Back–Translation (BT). In DAE, a sentence including pseudo noises is corrected by the Encoder–Decoder model. It is expected that Transformer can acquire the representation ability for each of the source and target sides. In BT, an auxiliary translation model from the target to source side is also trained and pseudo source sentences are produced using this model. The translation model from the source to target side is trained using the produced source sentences and corresponding original target sentences. It is expected that Transformer can acquire the translation ability from the source to target side.

Unsupervised MT techniques do not require a parallel but a comparable corpus as training data. Therefore, we use comparable translated texts using Google Translation as the source side data. Specifically, we use a monolingual corpus as target-side data and a monolingual corpus written in another language

translated into a GEC target language as source side data.

This paper makes the GEC models using the synthetic comparable corpus and a little but well-annotated data. We used the CoNLL-2014 dataset, the JFLEG dataset and W&I+LOCNESS dataset for English GEC evaluation and the NLPCC-2018 dataset for Chinese GEC evaluation. We analyzed the difference between SMT and NMT for the error correction performance. As for the unsupervised SMT, we determined the difference in correction via the forward and backward refinement.

The main contributions of this study are as follows:

- We introduced and evaluated unsupervised methods based on MT for GEC to address the issue of lack of a learner corpus.
- We used a synthetic comparable corpus, the source side of which consists of translated texts using MT, for unsupervised methods.

The structure of this thesis is shown below. In Section 1, we show the proposal and contribution of this thesis. In Section 2, we describe the related works on unsupervised MT and GEC. In Section 3, we explain unsupervised GEC method and show how to make the synthetic comparable corpus. In Section 4, the experiment setup and results are shown. In Section 5, we discuss the experiment results. In Section 6, we describe the summary of this thesis.

目次

図目次		ix
第 1 章	はじめに	1
第 2 章	関連研究	4
2.1	教師なし機械翻訳	4
2.2	文法誤り訂正	7
2.2.1	教師あり文法誤り訂正	7
2.2.2	少量データに対する文法誤り訂正	7
第 3 章	教師なし文法誤り訂正	9
3.1	擬似コンパラブルコーパス作成	9
3.2	統計的機械翻訳を用いた教師なし文法誤り訂正	9
3.2.1	言語横断的なフレーズ単位分散表現の作成	10
3.2.2	類似度を用いたフレーズテーブルの作成	12
3.2.3	SMT システムの更新	13
3.3	ニューラル機械翻訳を用いた教師なし文法誤り訂正	14
第 4 章	文法誤り訂正実験	15
4.1	実験設定	15
	データセット	15
	評価尺度	17
	SMT の設定	17
	スペルチェッカー	18
	NMT の設定	18
	比較モデル	18
4.2	英語文法誤り訂正の結果	19
	学習者対訳コーパス学習モデルと教師なし手法モデルの比較	19
	擬似対訳コーパス学習モデルと教師なし手法モデルの比較	19

	教師なし統計的機械翻訳と教師なしニューラル機械翻訳の比較	21
	Back-Translation の有無	21
	スペルチェッカーの有無	21
	教師なし統計的機械翻訳の更新方法の違い	21
	ワークショップ提出システムとの比較	22
4.3	中国語文法誤り訂正の結果	22
第 5 章	分析	24
5.1	擬似コンパラブルコーパス作成について	24
5.2	統計的機械翻訳を用いた文法誤り訂正について	24
5.3	ニューラル機械翻訳を用いた文法誤り訂正について	25
5.4	統計的機械翻訳とニューラル機械翻訳の比較	26
第 6 章	おわりに	28
	謝辞	29
	参考文献	30
	発表リスト	35

図目次

2.1	Masked Lanugage Model の概要	5
2.2	Denoising Auto-Encoder の概要	5
2.3	Back-Translation の概要	6
3.1	擬似コンパラブルデータ作成に関する概要	10

第 1 章 はじめに

近年、自然言語処理の分野において、言語学習者支援タスクとして、文法誤り訂正 (GEC) に関する研究が盛んに行われている。GEC は、言語学習者の文法的に誤った文を入力とし、訂正されてより流暢な文を出力とする翻訳問題として取り組まれていることが多い。この問題設定の類似性から、機械翻訳 (MT) で研究されている手法を GEC に適用した研究が数多く存在している。また、MT の研究と同様に、GEC では大量の対訳コーパスや擬似データを使用することで性能が改善されることが知られている [18, 26, 44]。

GEC において一般に使用することが可能なデータとしては、言語学習者の相互添削サイトである Lang-8 から抽出したものが知られている [33]。このデータは、学習者の文とそれを訂正した文として対訳関係になっており、対訳コーパスとして使用可能である。Lang-8 から抽出したうち、もっともデータ量の多い言語は英語であるが、MT のデータ量が豊富な言語対と比べるとまだまだ少なく、英語以外の言語についてはさらに訓練データが少ない。そのため、この訓練データが少量であることが GEC の問題の一つとなっている。実際、学習者の文は Lang-8* や HiNative[†] などの言語学習者支援サイトから抽出することで、大量に獲得できるが、この誤り文に対応する訂正文を得ることは簡単ではない。Lang-8 に投稿されている訂正文は、本当に正しい訂正文であるのかは保証されておらず、正しい訂正文であることを保証するために、学習者の対象言語の母語話者が精査した場合、そのコストが莫大となってしまう。このように、誤り文と訂正文が対応した大規模対訳コーパスを作成することは困難である。

Bryant ら [9] は、この使用可能な学習者データが少量である問題に対して、使用可能な訓練データを制限した英語の GEC のワークショップ (BEA2019 Low Resource Track) を開催した。このワークショップでは、大規模単言語データに擬似誤りを付与して擬似対訳データを作成する手法などが提案された。

本研究では、主に MT で研究されている教師なし手法を GEC に取り入れることで、使用可能な学習者データが少ない状況での実験を行った。具体的には、統計的

*<http://lang-8.com>

[†]<http://hinative.com>

機械翻訳 (SMT) を中心とした教師なし手法 [2, 30] とニューラル機械翻訳 (NMT) を中心とした教師なし手法 [13] をそれぞれ用いて, GEC モデルを構築した.

SMT を用いた手法 [2, 30] は, 誤り側と訂正側のフレーズ分散表現を共有したものを使用して, 誤りフレーズとそれを訂正したフレーズの対応をとる. その後, このフレーズテーブルを 2 種類の更新方法を用いて修正していく.

NMT を用いた手法は, Transformer [43] と呼ばれる自己回帰型注意機構を中心とした Encoder-Decoder ネットワークを使用する. この教師なし手法 [13] は, 事前に誤り側と訂正側で言語知識を共有するように Masked Language Model (MLM) と呼ばれる, 人手のラベルを必要としない学習を行う手法である. その後, 誤り側, 訂正側について, Denoising Auto-Encoder (DAE) と Back-Translation (BT) の目的関数を最小化するよう同時に学習していく. DAE はある文に対して表層的なノイズを加えて, それを Encoder-Decoder で修正する学習方法である. これにより, Transformer は誤り側, 訂正側それぞれに対する表現能力が獲得されると考えられる. BT は誤り側の文, または訂正側の文に対して現在学習中の誤り訂正モデルまたは誤り生成モデルで推論して, 出力された訂正文, または誤り文を獲得し, その文対を用いて逆方向のモデルである, 誤り生成モデルまたは誤り訂正モデルを訓練する学習方法である. この BT により, 誤り文を訂正文に翻訳する能力が得られると期待される.

また, 教師なし MT の手法は, 対訳コーパスではなく, 誤り側の文集合と訂正側の文集合となる, コンパラブルコーパスが訓練データとして必要になる. そのため, 本研究は誤り文として MT の出力文を使用することで, 擬似コンパラブルコーパスを作成した. 具体的には, GEC が対象としている言語の単言語コーパスを訂正側として, 別の言語で書かれた単言語コーパスを対象言語に翻訳したデータを誤り側とした.

本研究では, 作成した擬似コンパラブルコーパスと少量の学習者データを使用して GEC モデルを構築し, その性能を評価する. 評価データには英語学習者支援タスクとして CoNLL-2014 test data, JFLEG test data, W&I+LOCNESS data, 中国語学習者支援タスクとして NLPC-2018 test data を使用する. SMT と NMT の違いで, 最終的な結果にどのような差が生じるのか考察した. また, SMT を用いた手法については, フレーズテーブルの更新手法による出力の差異に

ついて分析を行った。さらに、最も良い手法を BEA2019 Low Resource Track で提案された他の手法と比較した。

本研究の貢献は以下の通りである。

- 文法誤り訂正で利用できる訓練データが少量である問題に対して、機械翻訳で研究されている教師なし手法を導入、その性能を検証した。
- 教師なし手法を導入する際に必要となるコンパラブルコーパスに、機械翻訳の出力を誤り側としたデータを使用し、その効果を検証した。

第 2 章 関連研究

2.1 教師なし機械翻訳

近年, NMT, SMT それぞれにおいて, 教師なし手法の研究が盛んである. 特に, Artetxe ら [2] による SMT を用いた教師なし手法 (USMT) は, まず言語横断的なフレーズ embedding を獲得, この embedding を元にフレーズテーブルを作成する. 次にこのフレーズテーブルと, 単言語データから学習した言語モデルを既存の SMT の枠組みに導入したものを初期モデルとする. そしてこの初期モデルを元に, 単言語データから擬似データを作成して SMT モデルを洗練していく. また Lample ら [29] や Marie and Fujita [30], Artetxe ら [3] は, USMT の最終的なモデルから作成した擬似データを, 教師なし NMT の最終モデルに追加する, または NMT モデルの初期化に使用する有効性についても報告している.

NMT を用いた教師なし手法 (UNMT) として, Conneau and Lample [13] は, 大規模単言語データを使用した事前学習と人手で付与された教師データなしで学習可能な目的関数を用いる手法を提案した. 彼らは, 事前学習として MLM と呼ばれる学習を行う. この学習の概要図を図 2.1 に示す. 彼らは MT での実験の際, それぞれの言語の単言語コーパスの各文に対して図のようにランダムにマスク ([mask]) を行い, このマスクされた単語を当てるように Encoder を学習している. この MLM の学習は言語共通の単一 Encoder を用いて行い, 言語の区別は language embedding と呼ばれる言語タグで行う. この MLM で事前学習を行うことで, 対象言語の表現能力が高い Encoder を学習できることが Devlin ら [15] によって報告されている. MLM で学習された Encoder のパラメータを, Encoder と Decoder のパラメータ初期化に使用し, DAE と BT を学習する. DAE に関する概要図を図 2.2 に示す. DAE はそれぞれの言語の単言語データの各文に対して単語の置換, マスク, 削除を行い, Encoder-Decoder で元の文を当てるように学習するタスクである. この DAE を通して, Encoder-Decoder モデルは各言語に対する表現能力が獲得できると考えられる. BT は図 2.3 のように, 本来の翻訳方向とは逆側のモデルを利用して推論を行い, 単言語データから擬似的に教師データを作成し, 学習する手法である. 図 2.3 では順方向モデルを訓練する概要図だが, 逆

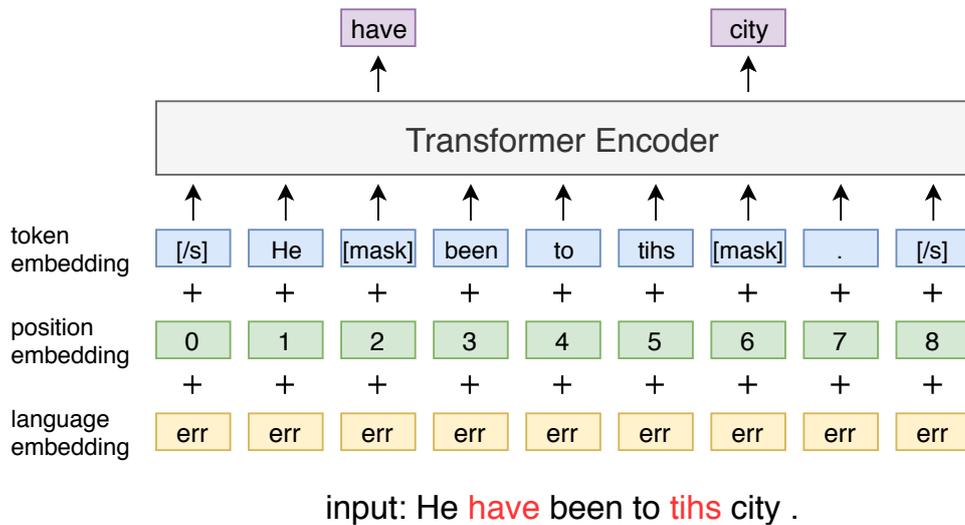


図 2.1 Masked Language Model の概要, 赤字は学習者の誤った箇所を示している.

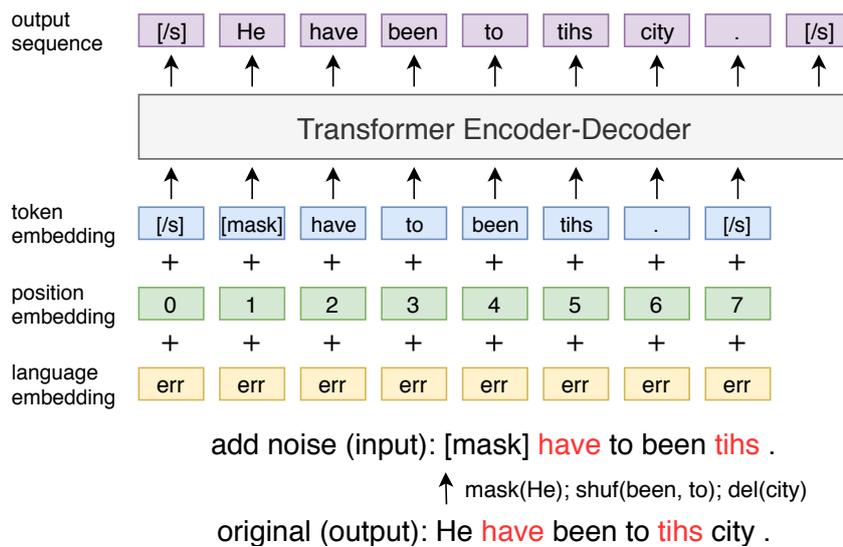
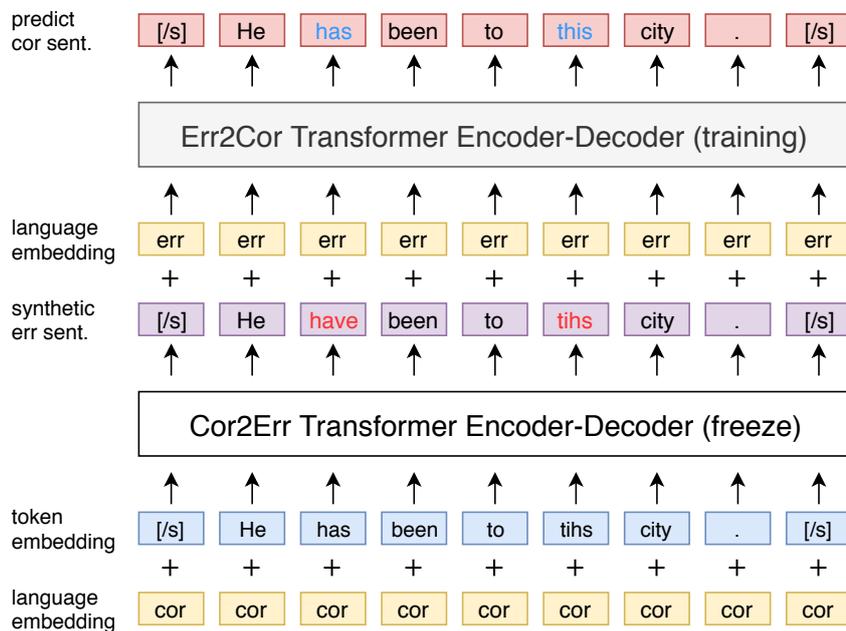


図 2.2 Denoising Auto-Encoder の概要, 赤字は学習者の誤った箇所を示している.

方向モデルも同様にして学習する. この BT を用いることで, Encoder-Decoder を用いた翻訳モデルを学習することが可能になる. BT を学習する際の逆方向モデルは順方向モデルと同一の Encoder-Decoder モデルを使用し, パラメータ共有を行っている. これら DAE と BT を用いることで UNMT の性能が向上すること



Cor sentence: He has been to this city .

図 2.3 Back-Translation の概要, position embedding は簡単のため省略している. また赤字は誤った箇所, 青字は訂正した箇所を示している.

がいくつかの研究で知られている [4, 28, 29]. また, Song ら [42] はこの MLM の事前学習を Encoder だけでなく, Encoder-Decoder の事前学習に拡張することで, Conneau and Lample [13] と比べて数ポイント翻訳性能が向上することを報告している.

我々は, これら USMT 手法と UNMT 手法を GEC に適用し, 性能調査を行った. 具体的には USMT 手法として, Artetxe ら [2] の手法を用いた. また UNMT として, ニューラル手法が事前学習と明示的な教師データを使用しない学習が GEC に有効なのかを調べるため, Conneau and Lample [13] の手法を GEC に適用した.

2.2 文法誤り訂正

2.2.1 教師あり文法誤り訂正

GEC の分野では，MT で研究された手法を取り入れた研究が多く存在する．Junczys-Dowmunt and Grundkiewicz [24] によって，SMT を GEC に用いることで高い訂正性能を得ることができると報告されている．この研究は SMT の代表的ツールである Moses [27] を用いて GEC モデルを作成している．MT との違いとして，素性に編集距離などの GEC 特有のものを使用している．この SMT を用いた GEC モデルは，RNN を用いた Encoder-Decoder モデル [45] と比べて訂正性能が高い．Chollampatt and Ng [12] は CNN を用いた Encoder-Decoder モデルを用いることで，この SMT モデルより高い性能が得られることを示した．さらに，Junczys-Dowmunt ら [25] は自己注意機構を用いた Encoder-Decoder モデルである Transformer [43] を用いることで，この CNN モデルと比べてより高い訂正性能を得ることができると示した．本研究では，SMT を中心とした手法と NMT モデル (Transformer) に基づく手法を用いた．

また近年，擬似データを含めた大規模コーパスを用いた教師あり NMT 手法を GEC に取り入れる研究が盛んである．現在英語の GEC で最高精度を報告している Kiyono ら [26] は，純粋な対訳データとして 561K 文対，擬似データとして 70M 文対のコーパスを使用し，教師あり NMT として Transformer [43] を学習している．しかしながら，他の言語の GEC について考えた際に，対訳コーパスとしてこの規模のデータを用意することは必ずしも簡単ではない．

2.2.2 少量データに対する文法誤り訂正

少量の学習者データや，対応する訂正文が少ない状況での GEC の研究もいくつか存在する．Park and Levy [40] は学習者の記述したテキストを用いた Noisy-channel モデルに基づく GEC モデルを提案した．一方で本研究は，学習者の記述したテキストはほぼ必要なく，代わりに単言語コーパスを必要とする．Bryant and Briscoe [7] は少量の学習者データと対応した訂正文を用いて GEC モデルを構築した．彼らのモデルは単言語データで作成された言語モデルと，英語の学習者の誤り

傾向に基づく訂正候補を用いている。本研究は対象言語の学習者の誤り傾向に関する知識は必要としない。Miao [31] らは Metropolis-Hastings アルゴリズムを用いてサンプリングを行い、出力文生成を行っている。この手法は学習時に対訳コーパスは必要とせず、代わりに単言語データを必要とする。本研究の手法は、パラメータ決定などのために少量の対訳データを使用する。

また、近年では擬似データを用いることで、学習者データが少量である状況に取り組んでいる研究がいくつか存在する。Zhao ら [46] は DAE の枠組みで擬似データを作成することで、GEC モデルの性能が改善されることを報告している。この擬似データは、単言語データに対してランダムに単語を置換するなどの人工的な誤りを付与することで生成されている。Bryant ら [9] は、使用可能な訓練データを制限した英語 GEC のワークショップを開催した。このワークショップで最高精度を報告した Grundkiewicz ら [22] は、大規模な単言語データに対して擬似誤りを付与することで擬似訓練データを作成、GEC モデルを構築している。また、彼らの手法を他の言語の GEC に適応した場合でも、同様に性能が向上することが Náplava and Straka [34] によって報告されている。

第 3 章 教師なし文法誤り訂正

この章では、MT で研究されている教師なし手法を GEC に適応する方法について述べる。3.1 章では教師なし手法が訓練データとして使用するための擬似コンパラブルデータの作成手法について述べる。3.2 章では SMT を用いた教師なし GEC について述べ、3.3 章では NMT を用いた教師なし手法の GEC への適応方法について述べる。

3.1 擬似コンパラブルコーパス作成

本研究で使用する教師なし手法は訓練データとして学習者の文集合と流暢な文集合のコンパラブルコーパスを使用する。実際、Artetxe ら [2] や Conneau and Lample [13] は訓練データとしてそれぞれの言語の単言語コーパスを使用していた。

本研究では学習者の記述した文を得ることが難しい状況で実験を行っている。この問題設定のために、目的言語側として誤り側の文集合を作成する必要がある。本研究では擬似学習者文として既存の翻訳システムの出力を使用する。具体的には、GEC の対象言語の単言語コーパスを流暢な文集合として使用し、同じドメインの別言語で記述された単言語コーパスを翻訳したものを擬似学習者の文集合とした。図 3.1 に概要図を示す。5.1 章で、擬似学習者の文を作成する際の元の言語 (Language B) による GEC モデルの性能への影響について分析を行った。

この翻訳システムを使用した擬似学習文には綴りの誤った単語は含まれない。そのため、本研究ではスペル訂正システムを前処理としてパイプライン的に実行した。具体的には、評価時の入力文に対してスペル訂正を行い、その出力を学習した誤り訂正モデルに入力している。

3.2 統計的機械翻訳を用いた教師なし文法誤り訂正

アルゴリズム 1 に本研究における SMT を用いた教師なし GEC 手法の疑似コードを記述する。大部分は Artetxe ら [2] の USMT 手法を元に行っている。

最初に、各単言語コーパスからフレーズ単位の分散表現を学習し、これらの分散

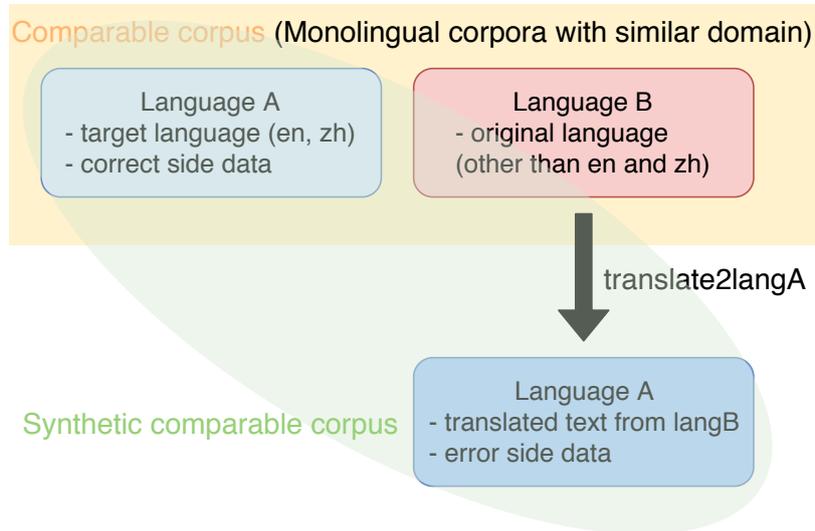


図 3.1 擬似コンパラブルデータ作成に関する概要

表現を共有空間へ埋め込むよう学習する。次に、この言語横断的な分散表現を元に、ベクトルの類似度を用いてフレーズテーブルを作成する。このフレーズテーブルと各単言語データを用いて学習した N-gram 言語モデルを用いてフレーズベースの SMT システムを作成する。最後に、この SMT システムに対して、順方向または逆方向の翻訳を用いて更新を行っていく。

3.2.1 言語横断的なフレーズ単位分散表現の作成

原言語側、目的言語側のそれぞれから N-gram embedding を作成する。具体的にはそれぞれの単言語データにおいて頻度の高い unigram, bigram, trigram* に対して skip-gram [32] の枠組みを用いて単言語分散表現を作成する。その後、作成した個々の単言語分散表現を共有言語横断空間へマッピングする。このマッピングに

*それぞれ学習データ内の頻度順に 200K, 400K, 400K を使用。

Algorithm 1 教師なし統計的機械翻訳を用いた文法誤り訂正モデルの学習

Require: 原言語側言語モデル LM_s
Require: 目的言語側言語モデル LM_t
Require: 原言語側単言語コーパス C_s
Require: 目的言語側単言語コーパス C_t
Require: 開発データ (少量のラベルデータ) T
Require: 繰り返し数 N
Ensure: 原言語 \rightarrow 目的言語フレーズテーブル $P_{s \rightarrow t}^{(N)}$
Ensure: 原言語 \rightarrow 目的言語の素性重み $W_{s \rightarrow t}^{(N)}$

- 1: $W_s^{emb} \leftarrow \text{TRAIN}(C_s)$
- 2: $W_t^{emb} \leftarrow \text{TRAIN}(C_t)$
- 3: $W_s^{cross_emb}, W_t^{cross_emb} \leftarrow \text{MAPPING}(W_s^{emb}, W_t^{emb})$
- 4: **if** forward_refinement **then**
- 5: $P_{s \rightarrow t}^{(0)} \leftarrow \text{INITIALIZE}(W_s^{cross_emb}, W_t^{cross_emb})$
- 6: $W_{s \rightarrow t}^{(0)} \leftarrow \text{TUNE}(P_{s \rightarrow t}^{(0)}, LM_t, T)$
- 7: **for** $iter = 1, \dots, N$ **do**
- 8: $\text{synthetic_data}_t \leftarrow \text{DECODE}(P_{s \rightarrow t}^{(iter-1)}, LM_t, W_{s \rightarrow t}^{(iter-1)}, C_s)$
- 9: $P_{s \rightarrow t}^{(iter)} \leftarrow \text{TRAIN}(C_s, \text{synthetic_data}_t)$
- 10: $W_{s \rightarrow t}^{(iter)} \leftarrow \text{TUNE}(P_{s \rightarrow t}^{(iter)}, LM_t, T)$
- 11: **else if** backward_refinement **then**
- 12: $P_{t \rightarrow s}^{(0)} \leftarrow \text{INITIALIZE}(W_t^{cross_emb}, W_s^{cross_emb})$
- 13: $W_{t \rightarrow s}^{(0)} \leftarrow \text{TUNE}(P_{t \rightarrow s}^{(0)}, LM_s, T)$
- 14: **for** $iter = 1, \dots, N$ **do**
- 15: $\text{synthetic_data}_s \leftarrow \text{DECODE}(P_{t \rightarrow s}^{(iter-1)}, LM_s, W_{t \rightarrow s}^{(iter-1)}, C_t)$
- 16: $P_{s \rightarrow t}^{(iter)} \leftarrow \text{TRAIN}(\text{synthetic_data}_s, C_t)$
- 17: $W_{s \rightarrow t}^{(iter)} \leftarrow \text{TUNE}(P_{s \rightarrow t}^{(iter)}, LM_t, T)$
- 18: $\text{synthetic_data}_t \leftarrow \text{DECODE}(P_{s \rightarrow t}^{(iter-1)}, LM_t, W_{s \rightarrow t}^{(iter-1)}, C_s)$
- 19: $P_{t \rightarrow s}^{(iter)} \leftarrow \text{TRAIN}(\text{synthetic_data}_t, C_s)$
- 20: $W_{t \rightarrow s}^{(iter)} \leftarrow \text{TUNE}(P_{t \rightarrow s}^{(iter)}, LM_s, T)$

は Artetxe ら [1] の手法を使用し、教師なしでマッピングを行っている。

3.2.2 類似度を用いたフレーズテーブルの作成

作成した言語横断的な N-gram embedding からフレーズテーブルを作成する。具体的にはフレーズ翻訳モデルと語彙翻訳モデルを作成する。

原言語側のフレーズ \bar{e} に対する目的言語側のフレーズ \bar{f} のフレーズ翻訳モデル $\phi(\bar{f}|\bar{e})$ は、ある原言語側のフレーズに対して、共有言語横断空間内の 100 近傍の目的言語のフレーズを候補とした。フレーズ翻訳モデルのスコアはある原言語側のフレーズと目的言語側のフレーズのコサイン類似度を正規化したものを使用している。具体的には次式の通りである。

$$\phi(\bar{f}|\bar{e}) = \frac{\exp(\cos(\bar{e}, \bar{f})/\tau)}{\sum_{\bar{f}'} \exp(\cos(\bar{e}, \bar{f}')/\tau)}$$

\bar{f}' は目的言語側のフレーズ集合の各要素を表しており、 τ は予測の信頼度を制御する温度パラメータである。 τ は Artetxe らと同様に、ある embedding に対して逆側の最近傍のフレーズの embedding と、元の embedding のフレーズ翻訳確率が最大になるように推定する。例えば原言語側の単語として “teh” があり、目的言語側の最近傍単語が “the” だった場合、 $\max_{\tau} \log \phi(\text{the}|\text{teh})$ のように、フレーズ翻訳確率が最大となるように推定している。また、逆側のフレーズ翻訳確率についても同じ τ を使用しており、実際には以下のように同時に学習している。

$$\max_{\tau} \left(\sum_{\bar{f}} \log \phi(\bar{f}|\text{NN}_{\bar{e}}(\bar{f})) + \sum_{\bar{e}} \log \phi(\bar{e}|\text{NN}_{\bar{f}}(\bar{e})) \right)$$

ただし、 $\text{NN}_{\bar{e}}(\bar{f})$ は目的言語側のフレーズに対して原言語側で最近傍のフレーズである。目的言語側のフレーズに対する原言語側のフレーズのフレーズ翻訳モデル $\phi(\bar{e}|\bar{f})$ も同様に構築する。

原言語側のフレーズ \bar{e} に対する目的言語側のフレーズ \bar{f} の語彙翻訳モデル $\text{lex}(\bar{f}|\bar{e})$ は、原言語側のフレーズ内の各単語に対して、目的言語側のフレーズ内で翻訳確率が最も高い単語を対応した単語とする。つまり語彙翻訳モデルのスコアは原言語側のフレーズ内の各単語に対応する翻訳確率の積を用いる。具体的には次式の通りである。

$$\text{lex}(\bar{f}|\bar{e}) = \prod_i \max \left(\epsilon, \max_j \phi(\bar{f}_i|\bar{e}_j) \right)$$

ϵ は対応する単語が存在しない場合のための定数項である。本研究では Artetxe らと同様に 0.001 とした。目的言語側のフレーズに対する原言語側のフレーズの語彙翻訳モデル $\text{lex}(\bar{e}|\bar{f})$ も同様である。

3.2.3 SMT システムの更新

上記で作成したフレーズテーブルは trigram までしか考慮されておらず、フレーズ翻訳モデル、語彙翻訳モデルそれぞれのスコアも言語横断的な embedding を元に推定したものである。そのため、得られたフレーズテーブルは言語横断的 embedding の埋め込み精度に強く依存しており、誤ったフレーズ対応も多く含まれると考えられる。本研究では、SMT システムを用いてフレーズテーブルの更新を行い、この問題に対応する。具体的には、言語モデルを使用した SMT システムを用いて推論を行い、得られた出力と元の単言語データを利用して SMT システムの更新を行う。言語モデルを用いることで、より流暢な出力が得られ、誤ったフレーズ対応も軽減できると期待される。この更新処理は Algorithm 1 の 4-20 行目に対応している。SMT システムの更新手法として forward refinement と backward refinement の 2 種類の手法を用いた。

forward refinement では、順方向の推論結果を使用して SMT システムの更新を行う。初期フレーズテーブル $P_{s \rightarrow t}^{(0)}$ と目的言語側の言語モデル LM_t を用いて、原言語側の単言語コーパス C_s に対する目的言語側の擬似データ synthetic data_t を作成する。この目的言語側が推論結果である擬似対訳データを用いて SMT を学習、 $P_{s \rightarrow t}^{(1)}$ を作成する。またこの時の SMT システムの素性の重みを少量のラベルデータ T を用いて決定する。この操作を繰り返し数 N だけ実行する。

backward refinement では、逆翻訳の機構を利用することでフレーズテーブルの更新を行う。初期フレーズテーブル $P_{t \rightarrow s}^{(0)}$ と言語モデル LM_s を用いて、目的言語側の単言語コーパス C_t に対する原言語側の擬似データ synthetic data_s を作成する。この原言語側が擬似データとなっている擬似対訳データを用いて SMT を学習、 $P_{s \rightarrow t}^{(1)}$ を作成する。この $P_{s \rightarrow t}^{(1)}$ を用いて、原言語側の単言語データ C_s を翻訳、目的言語側が擬似的になっている擬似対訳データ synthetic data_t を作成する。この擬似データを用いて $P_{t \rightarrow s}^{(1)}$ を学習する。また $P_{s \rightarrow t}^{(1)}$ や $P_{t \rightarrow s}^{(1)}$ を更新した後、SMT

システムの素性の重みを少量のラベルデータ T に対して決定している。この操作を予め決めた繰り返し数 N だけ実行する。

3.3 ニューラル機械翻訳を用いた教師なし文法誤り訂正

本研究では Conneau and Lample [13] と同様に MLM で Encoder を学習した後、DAE や BT を用いて Encoder-Decoder を学習している。MLM の学習は図 2.1 のように、各単言語コーパスの文に対してマスクを入れ、マスクした単語を当ててるように学習する。この時、language embedding として学習者側を表す err と訂正側 cor を使用し、単一の Encoder を学習する。この学習した Encoder のパラメータを Transformer の Encoder と Decoder のパラメータの初期化に使用する。

DAE は Conneau and Lample と同様に各単言語コーパスの文に対してランダムに単語置換とマスク、単語削除を行い、Encoder-Decoder を用いて元の文を再現するよう学習している。ただし、GEC は誤った文を訂正するタスクであるため、Conneau and Lample と比べてノイズの度合いを強めている。また、BT も Conneau and Lample と同様に行っている。図 2.3 のように、誤り生成モデルを用いて訂正文側の単言語コーパスに対して推論を行い、擬似誤り文を生成している。この擬似誤り文と訂正文側の単言語コーパスで構築される擬似対訳データを用いて誤り訂正モデルを学習している。また、誤り生成モデルについても同様に学習を行っている。

本研究では訂正モデルとして、Encoder-Decoder を DAE と BT を同時に学習したモデルと DAE のみで学習したモデルを用いた。DAE のみで学習したモデルは、Encoder-Decoder 学習時に入力にノイズの加えられた文で、それを訂正するよう学習するため、推論時の language embedding には cor を使用した。DAE と BT で学習したモデルは、推論時の language embedding には err を使用している。また DAE と BT で同時学習する際は、それぞれの loss に重みをかけた。

第 4 章 文法誤り訂正実験

4.1 実験設定

表 4.1 文法誤り訂正で使した訓練データと開発データ.

Corpus	文数	学習者データ
En GEC 訓練データ		
Finnish News Crawl	1,904,880	No
English News Crawl	2,116,249	No
One Billion Word Benchmark	24,482,651	No
Zh GEC 訓練データ		
English Wikipedia	2,798,555	No
Chinese Wikipedia for comparable	2,888,276	No
Chinese Wikipedia for LM	10,167,366	No
for CoNLL-14 (En)		
CoNLL-13	1,312	Yes
for JFLEG test (En)		
JFLEG dev	754	Yes
for W&I+LOCNESS (En)		
tuning data	2,191	Yes
dev data	2,193	Yes
for NLPCC-2018 (Zh)		
Lang-8 data for dev	2,000	Yes

■**データセット** 本研究では，教師なし GEC の性能検証として英語の GEC タスクと中国語 GEC タスクを行った．中国語 GEC タスクについては Mandarin 簡体字を対象としている．表 4.1 に GEC モデルの学習に使した訓練データと開発データを示す．英語の GEC について，明記しない限り Finnish News Crawl 2014–2015 の一部を英語に翻訳したものを誤り側とし，English News Crawl 2017 の一部を訂正側の訓練データとして用いている．訂正側の言語モデル LM_t として，English News Crawl で訓練したものに追加して，One Billion Word Benchmark [11] データで訓練したものも用いている．中国語の GEC については，English

Wikipedia の一部を中国語に翻訳したものを誤り側とし、Chinese Wikipedia の一部を訂正側の訓練データとして用いている。さらに追加の言語モデルとして Chinese Wikipedia 全体で学習したものも用いている。また擬似誤り作成時の翻訳モデルとして、Google 翻訳を `googletrans v2.4.0` を通して用いた。

本研究では英語 GEC の評価データとして CoNLL-14 [37] と JFLEG test set [36], W&I+LOCNESS [9, 20] を用いた。また、CoNLL-13 data set を CoNLL-14 のベストモデル決定や SMT の素性の重みの決定に、JFLEG dev set を JFLEG test set のベストモデル決定や SMT の重み決定に用いた。ただし、UNMT の手法については、パラメータ探索に CoNLL-14 も用いており、評価として JFLEG test set と W&I+LOCNESS を用いている。また、Bryant ら [9] の開催したワークショップでは、学習者データとして W&I+LOCNESS の開発データのみ使用が許された。そのため、本研究ではこれを半分に分割し^{*}, tune data と dev data として、tune data はベストモデル決定や SMT の重み決定に、dev data は評価データとして用いた。また中国語 GEC の評価データには NLPCC-2018 の評価データを用いた。開発データには Lang-8 から抽出した中国語学習者の文とそれを訂正した 2K 文対 (Lang-8 data for dev) を使用した。

英語 GEC である CoNLL-14 と JFLEG test set について、単語分割は NLTK [5] を用いた。W&I+LOCNESS については `spaCy v1.9.0†` と `en_core_web_sm-1.2.0` を単語分割に用いた。また English News Crawl で学習した `moses truecaser‡` を用いて使用する英語データの truecase を行った。これらの処理を行ったデータに対して English News Crawl でオペレーション数 50K の byte-pair-encoding (BPE) [41] を行い、英語データの subword 化を行った。

中国語 GEC である NLPCC-2018 については SMT を用いた場合と NMT を用いた場合で単語分割を変えている。SMT を用いた場合は PKUNLP[§] を用いて単語分割を行い、その後 BPE を用いて全ての単語に対して subword 化を行った。NMT を用いた場合は、中国語は文字分割を行い、それ以外の単語はそのまま元の

^{*}W&I+LOCNESS では学習者のレベルが 4 段階に分かれているため、分割する際に等しくなるようにした。

[†]<https://github.com/explosion/spaCy>

[‡]<https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/truecase.perl>

[§]<http://www.icst.pku.edu.cn/lcwm/pkunlp/downloads/libgrass-ui.tar.gz>

空白区切りを行った。この中国語かどうかの判定には Unicode を用いた。その後 BPE を行い、中国語以外の単語について subword 化を行った。BPE のオペレーション数は 30K である。評価する際は、出力文について detokenize を行い、その後 PKUNLP を用いて単語分割を行ったものを gold と比較した。

■**評価尺度** 本研究では CoNLL-14, W&I+LOCNESS dev, NLPCC-2018 test set について Precision と Recall, $F_{0.5}$ を報告する。CoNLL-14 と NLPCC-2018 の各スコアは M² Scorer [14] を用いて算出し、W&I+LOCNESS dev のスコアは ERRANT scorer [8] を用いて求めた。JFLEG test set に対しては GLEU [35] を用いた GLEU スコアを報告する。

■**SMT の設定** USMT の実装については Artetxe ら [2] の実装[¶] を修正したものを使用した。具体的には、以下の素性を追加した。

- 単語単位の編集距離
- 単語単位, 文字単位の編集操作 (置換, 削除, 挿入)
- operation sequence model^{||} [16]
- 9-gram 単語クラス言語モデル**

これらの素性は Grundkiewicz and Junczys-Dowmunt [21] で使用された素性からスパース素性を除いたものである^{††}。また、単語クラス言語モデルは 英語の場合 One Billion Word Benchmark データを用いて学習し、中国語の場合 Chinese Wikipedia 全体を用いて学習した。この時のクラス数はどちらも 200 とし、単語クラスは fastText [6] を用いて推定した。

SMT システム自体は Moses [27] を使用した。アライメントツールとして FastAlign [17] を使用し、KenLM [23] を用いて 5-gram 言語モデルを作成した。また、素性の重み決定には MERT [38] を用いており、CoNLL-14 には CoNLL-13 に対して $F_{0.5}$ の値が最大になるようにした。また W&I+LOCNESS dev データ

[¶]<https://github.com/artetxem/monoses>

^{||}OSM 素性は更新処理の時にのみ使用した。

**単語クラス言語モデルは誤り生成モデルの学習時には使用していない。

^{††}本研究で使用した開発データは Grundkiewicz and Junczys-Dowmunt [21] が使用したものより小さく、スパース素性がより疎になってしまうため使用していない。

には tune データに対して $F_{0.5}$ が最大になるように, JFLEG test には dev データに対して GLEU スコアが最大になるように決定した. NLPCC-2018 test データには Lang-8 から抽出した開発データに対して $F_{0.5}$ が最大になるようにした. SMT システム更新時の擬似文対に対して, [3, 80] の文長制限を行った. 繰り返し数 N は 3 を使用し, フレーズ分散表現は 300 次元とした.

■**スペルチェッカー** また, 英語 GEC については明記しない限り, スペルチェッカーとして `pyspellchecker`^{††}を用いている. これは編集距離を用いて, ある単語に対して単語リスト内に存在する編集距離 2 の候補を提示するツールである. 本研究では One Billion Word Benchmark 内に 5 回以上出現する単語から単語リストを作成している. 中国語 GEC についてはスペルチェッカーは用いていない.

■**NMT の設定** また, UNMT の実装は Conneau and Lample [13] の実装^{§§}を使用した. ハイパーパラメータに関しては, CoNLL-14 に対する結果から DAE のノイズとして単語削除と単語マスクは 0.3, シャッフルは 5 単語以内で行うこととした. また DAE のみで Encoder-Decoder を学習する場合は初期学習率は $1e-4$, dropout は 0.1 とした. DAE と BT を同時学習する際は初期学習率は $3e-4$, dropout は 0.3 とし, DAE と BT の loss の比を 1:0.01 とした. また, ベストモデルには各開発データに対してパープレキシティが最小となるモデルを使用した. 明記していないハイパーパラメータについては元実装の値をそのまま使用した.

■**比較モデル** 比較のため, 英語 GEC について Lang-8 [33] から抽出した学習者文と対応する訂正文の対訳コーパス約 1.3M 文対を用いて SMT と NMT それぞれ学習したモデル (Lang-8 train) についても評価実験を行った. また, Zhao ら [46] の擬似データ作成手法を訂正側の単言語コーパスに適応し, 作成した擬似対訳コーパスを用いて訓練を行ったモデル (synthetic train) についても評価実験を行った. 教師あり SMT について, 使用した素性や実験設定は教師なしのものと同様である. また教師あり NMT については, fairseq [39] の CNN Encoder-Decoder [19] モデルを実験した. CNN Encoder-Decoder のハイパーパラメータについては Ge ら [18] の値と同じものを使用した.

^{††}<https://github.com/barrust/pyspellchecker>

^{§§}<https://github.com/facebookresearch/XLM>

他の関連研究のシステムとの比較として, Bryant ら [9] の開催したワークショップ (BEA2019 Low Resource Track) に提出されたシステムと比較を行った. このとき, 我々はこのワークショップへの提出システムとして W&I+LOCNESS dev に対して最も良いシステムを用いた.

4.2 英語文法誤り訂正の結果

■**学習者対訳コーパス学習モデルと教師なし手法モデルの比較** 表 4.2 に本研究で行った英語 GEC 実験の結果を示す. CoNLL-14 と JFLEG の結果から, USMT について forward refinement の方が訂正性能が高いと判断し, W&I+LOCNESS dev に対してはこちらの更新手法のみ使用している. CoNLL-14 では, $USMT_{\text{forward}} \text{ iter } 1$ は学習者データである Lang-8 で訓練した教師あり SMT と比べると $F_{0.5}$ が 13.57 ポイント低く, 教師あり NMT と比べると 17.17 ポイント低い. JFLEG でも, USMT の中では $USMT_{\text{forward}} \text{ iter } 1$ が最も性能が高いが, Lang-8 で訓練した教師あり SMT と比べて GLEU スコアが 5.28 ポイント低く, 教師あり NMT と比べると 3.39 ポイント低い.

■**擬似対訳コーパス学習モデルと教師なし手法モデルの比較** また, 表 4.2 の教師あり SMT の synthetic train と $USMT_{\text{forward}} \text{ iter } 1$ を比較すると CoNLL-14 では $F_{0.5}$ が 3.49 ポイント提案手法が向上していることがわかる. 一方で, 他の評価データに対しては synthetic train の方が訂正性能が高いことがわかる. 特に W&I+LOCNESS に対しては synthetic train の方が $F_{0.5}$ が 3.97 ポイント高いことがわかる. このように SMT モデル内では評価データによってどちらが訂正精度が良いか変わる. CoNLL-14 は入力として習熟度の高い学習者の文が用意されているのに対して, W&I+LOCNESS は様々な習熟度の学習者の文が入力となっている. そのため, 提案手法である擬似コンパラブルコーパスを用いた訂正モデルは, 習熟度の高い学習者の文を訂正する性能が擬似対訳コーパスを用いた場合と比べて高いと考えられる. また $USMT_{\text{forward}} \text{ iter } 1$ と 擬似対訳コーパスを用いて学習した NMT を比較した場合, CoNLL-14 と W&I+LOCNESS については USMT の方が優れているが, JFLEG に対しては擬似対訳コーパスを用いて学習した NMT の方が GLEU スコアが高いことがわかる. これらのことから, 少量のラベルデー

表 4.2 英語 GEC の実験結果. 太字が教師なし手法の中で最も良い結果を, 下線部が全体の中で最も良い結果を示す. UNMT は CoNLL-14 を用いてハイパーパラメータを決定したため, *は探索したハイパーパラメータの中での上限を示す.

	iter	CoNLL-14 (M ²)			W&I+LOCNESS dev			JFLEG
		P	R	F _{0.5}	P	R	F _{0.5}	GLEU
No edit	-	-	-	-	-	-	-	40.54
spellchecker	-	45.12	4.65	16.47	<u>41.52</u>	4.40	15.46	47.30
Supervised NMT								
Lang-8 train	-	<u>53.11</u>	26.47	<u>44.21</u>	-	-	-	54.04
synthetic train	-	31.95	15.01	26.06	15.98	10.04	14.29	50.90
Supervised SMT								
Lang-8 train	-	43.02	33.18	40.61	-	-	-	<u>55.93</u>
synthetic train	-	26.61	16.14	23.55	27.34	16.08	<u>23.98</u>	51.58
Unsupervised SMT								
USMT _{forward}	0	21.82	36.75	23.75	14.60	19.44	15.37	49.94
w/ forward_refine	1	25.92	32.65	27.04	20.58	18.04	20.01	50.65
	2	25.58	31.02	26.51	20.34	17.56	19.71	50.19
	3	23.95	33.13	24.54	18.86	17.56	18.58	50.40
w/o spellchecker	1	23.90	29.94	24.91	17.59	14.63	16.91	46.50
w/ backward_refine (USMT _{backward})								
	1	22.39	33.39	23.97	-	-	-	49.02
	2	24.96	27.13	25.36	-	-	-	48.90
	3	26.07	21.01	24.87	-	-	-	48.75
Unsupervised NMT (UNMT)								
w/ DAE	-	19.20*	18.69*	19.10*	8.74	10.68	9.07	45.38
w/o spellchecker	-	15.54*	16.98*	15.81*	6.41	7.65	6.62	41.14
w/ DAE, BT	-	20.13*	5.82*	13.49*	10.63	3.95	7.94	46.91
w/o spellchecker	-	6.96*	1.49*	4.01*	1.87	0.54	1.25	40.82

タが利用可能な状況では, 学習者の習熟度の高い入力に対しては本研究の手法が有効であるが, そうでない場合については擬似対訳データを用いて訂正モデルを訓練した方が良いことがわかる.

■**教師なし統計的機械翻訳と教師なしニューラル機械翻訳の比較** USMT と UNMT を比較すると、それぞれで一番良い手法の差が CoNLL-14 の場合 $F_{0.5}$ が 7.94 ポイントで、W&I+LOCNESS は 10.94 ポイント、JFLEG は GLEU が 5.27 である。この結果から、全ての評価尺度で USMT が優れていることがわかる。

■**Back-Translation の有無** UNMT は Encoder-Decoder を DAE のみで学習した場合と、DAE と BT で同時に学習した場合の結果を記載している。この結果から DAE と BT で同時に学習を行う場合よりも、DAE のみで学習した方が訂正精度が高いことがわかり、これは Conneau and Lample [13] の報告している機械翻訳の結果とは異なる。5.3 章で DAE と BT の組み合わせについて議論を行う。

■**スペルチェッカーの有無** また、USMT と UNMT について、前処理でスペルチェッカーを使用せずに実験を行った結果も表 4.2 に記載している。USMT_{forward iter 1} はスペルチェッカーの有無による訂正精度の差が CoNLL-14 では 2.13 ポイント、W&I+LOCNESS では 3.10 ポイント、JFLEG では 4.15 ポイントであった。UNMT では、DAE のみで学習した場合は USMT_{forward iter 1} と同程度のスコア差であることがわかる。一方で、DAE と BT を同時学習した場合においてスペルチェッカーの有無は大きく訂正精度に影響していることがこの表からわかる。これは DAE と BT を同時学習したモデル自体はあまり訂正を行わず、前処理として実行したスペルチェッカーの精度が支配的になっているからであると考えられる。実際に DAE と BT を同時学習したモデルについて、スペルチェッカーを除いた訂正モデルが CoNLL-14 に対して提案した訂正数を数えたところ、606 回あった。一方で、DAE のみ学習したモデルは 2,593 回、USMT_{forward iter 1} は 3,351 回となり、DAE と BT で学習したモデルの訂正は消極的であることがわかった。

■**教師なし統計的機械翻訳の更新方法の違い** 分散表現を用いたフレーズテーブルから 1 回目の更新が最も精度が高く、特に forward refinement の方が向上している。しかしながら、その後の forward refinement を用いた更新では性能が下がっていることがわかる。この更新手法では、学習者文側のデータは常に同じものを使用し、訂正文側が各更新毎に生成されている。このことから、学習者文側のデータの質は更新手法で特に重要であると考えられるが、本研究ではこのデータとして機械翻訳の出力文を利用している。そのため、学習者文側のデータの質はそれほど高

表 4.3 BEA2019 Low Resource Track に提出された他モデルとの比較. TMU が本研究の USMT_{forward} iter 1 である.

Team	TP	FP	FN	P	R	F _{0.5}
UEDIN-MS	2,312	982	2,506	70.19	47.99	64.24
Kakao&Brain	2,412	1,413	2,797	63.06	46.30	58.80
LAIX	1,443	884	3,175	62.01	31.25	51.81
CAMB-CUED	1,814	1,450	2,956	55.58	38.03	50.88
UFAL, Charles University, Prague	1,245	1,222	2,993	50.47	29.38	44.13
Siteimprove	1,299	1,619	3,199	44.52	28.88	40.17
WebSpellChecker.com	2,363	3,719	3,031	38.85	43.81	39.75
TMU	1,638	4,314	3,486	27.52	31.97	28.31
Buffalo	446	1,243	3,556	26.41	11.14	20.73

くなく、1 回目以降の更新では性能が改善していないと考えられる。

■ワークショップ提出システムとの比較 表 4.3 に BEA2019 Low Resource Track に提出された他の手法との比較を示す。TP, FP, FN はそれぞれ True Positive, False Positive, False Negative を意味している。表 4.2 の結果から、W&I+LOCNESS dev に対して最も訂正精度の高い USMT_{forward} iter 1 を我々は使用した。上位のシステムである UEDIN-MS や Kakao, LAIX は全て単言語コーパスに対して擬似誤りを加えることで擬似対訳コーパスを作成し、NMT モデルを学習した手法である。特に UEDIN-MS は擬似対訳コーパスとして 100M 文対使用している。一方で本研究の擬似対訳データを用いた NMT モデルはそれほど訂正精度が高くない結果となった。このことから、擬似対訳データを使用して NMT モデルを訓練する場合は特に多くのデータ量を必要とすると考えられる。また、そのような大規模擬似対訳データを用いて訓練したモデルと本研究の提案手法の訂正精度には大きく差があることがわかる。

4.3 中国語文法誤り訂正の結果

表 4.4 に中国語 GEC の結果を示す。提案手法である教師なし手法の中で最も良かったのは USMT_{forward} iter 3 であった。英語の場合は iter 1 が最も良かったが、

表 4.4 中国語 GEC の実験結果. 太字が教師なし手法の中で最も良い結果を, 下線部が全体の中で最も良い結果を示す.

	iter	NLPCC-2018 (M^2)		
		P	R	$F_{0.5}$
Supervised NMT				
synthetic train	-	<u>13.99</u>	6.95	11.64
Supervised SMT				
synthetic train	-	8.24	12.97	8.89
Unsupervised SMT				
w/ forward_refine	0	9.45	14.82	10.18
	1	10.28	16.72	11.14
	2	10.43	17.34	11.34
	3	11.21	17.51	12.08
w/ backward_refine				
	1	9.96	15.80	10.76
	2	8.54	12.26	9.10
	3	9.44	13.18	10.01
Unsupervised NMT				
w/ DAE	-	9.17	9.98	9.32
w/ DAE, BT	-	3.56	2.48	3.28

中国語の場合は更新を行うごとに訂正モデルの性能が向上していることがわかる.

この結果から, 擬似対訳データで学習した SMT は USMT と比較すると, 訂正精度が低いことがわかる. 擬似対訳データで学習した NMT は USMT と比較すると, $F_{0.5}$ で 0.44 ポイント USMT_{forward iter 1} の方が訂正精度が高いことがわかる. このように, 中国語 GEC では USMT と擬似対訳データを用いて学習した NMT が同程度の訂正性能であることがわかる.

また, UNMT と USMT を比較すると, 英語の場合と同様に USMT の方が訂正精度が高いことがわかる. この結果から, 現状の教師なし手法は言語によらず SMT に基づいた方法を用いた手法が GEC で有効であると考えられる.

第 5 章 分析

5.1 擬似コンパラブルコーパス作成について

表 5.1 擬似コンパラブルコーパス作成時の翻訳元言語ごとの訂正性能への影響。これらの単言語コーパスは News Crawl 2017 を使用している。使用した文数は約 20M で、これらの結果は USMT_{forward} iter 1 を用いている。

Src	Precision	Recall	F _{0.5}
Fi News Crawl	29.17	28.52	29.04
Ru News Crawl	27.11	29.84	27.62
Fr News Crawl	25.05	30.27	25.94
De News Crawl	23.26	26.04	25.04

本研究では、訓練データとして擬似学習者文を作成する際の翻訳元の言語と訂正モデルの性能の関係を調査した。表 5.1 は翻訳元の言語を変えた際の CoNLL-14 に対する USMT モデルの性能を示したものである。この表から、フィンランド語に対する翻訳結果を用いた場合が最も良い結果を得られることがわかる。そして英語により類似する言語 (Fr, De) については訂正精度が低くなっていることがわかる。このことから、誤り文として訂正対象の言語に対して類似していない言語を翻訳したものを使用することで、モデルの訂正性能が向上すると考えられる。これは、類似していない言語の方が機械翻訳も難しくなり、出力文に誤りが含まれやすくなるからであると考えられる。

5.2 統計的機械翻訳を用いた文法誤り訂正について

SMT システムの更新手法である forward refinement と backward refinement の違いを調べるためそれぞれの手法で更新されたモデルによって、CoNLL-14 データに対して提案された訂正の数を調査した。USMT_{forward} iter 1 と iter 2 はそれぞれ 3,437, 3,257 フレーズ訂正していた。一方で USMT_{backward} in iter 1 と iter 2 はそれぞれ 4,092, 2,789 フレーズ訂正していた。このように、USMT_{backward}

表 5.2 DAE と BT の loss の比率を変えた時の CoNLL-14 の訂正精度。学習したモデルの性能を調べるためスペルチェッカーは使用していない。

DAE	BT	Precision	Recall	F _{0.5}
1	1	3.47	1.15	2.47
1	0.1	9.17	1.00	3.47
1	0.01	9.31	1.14	3.83
1	0	15.54	16.98	15.81

では 2 回目の更新の結果, 1,303 回訂正しなくなっていることがわかる。Artetxe ら [2] や Lample ら [29] は backward refinement を用いた更新を行うことで教師なし MT の翻訳精度が向上することを報告している。GEC においては backward refinement を用いた更新によってシステムの訂正が消極的になることで訂正精度を改善していると考えられる。

表 4.2 の結果から, $USMT_{\text{backward}}$ と比較して, $USMT_{\text{forward}}$ の方が GLEU スコアが高いことがわかる。これは GLEU スコアは特に出力の流暢性を評価しており, forward refinement では擬似訂正文を作成する際に大規模言語モデルを考慮しているため, 流暢な擬似データを作成しているからであると考えられる。この流暢さを比較するため, JFLEG に対して最も良い更新モデルの出力を Common Crawl [10] で学習した言語モデル* に基づくパープレキシティで評価した。 $USMT_{\text{forward}}$ iter 1 の出力のパープレキシティは 179.23 である一方, $USMT_{\text{backward}}$ iter 1 に対するパープレキシティは 187.49 であった。この結果から $USMT_{\text{forward}}$ は Common Crawl で学習した言語モデルの元で $USMT_{\text{backward}}$ よりも流暢な訂正文を出力できていると考えられる。

5.3 ニューラル機械翻訳を用いた文法誤り訂正について

表 4.2 から, DAE と BT の同時学習ではなく, DAE のみを学習に用いた方が訂正精度が高いことがわかる。ここで我々は BT の影響を調べるため, DAE のみ

* <http://data.statmt.org/romang/gec-emnlp16/cclm.tgz>

で学習した訂正モデルに対して、BT の目的関数を最小化するように追加学習する実験を行った。この追加学習の際、DAE に対する BT の性能を調べるため、DAE の目的関数も最小化するようにしており、この DAE と BT の loss の重みを変えることで BT の影響を調査した。この分析ではスペルチェッカーは使用しておらず、loss の重み以外のパラメータは全て揃えている。

表 5.2 に DAE と BT の loss の比率を変えた場合の CoNLL-14 に対する実験結果を示す。この結果から、BT の loss の値は小さくするほど訂正精度が向上していることがわかる。また、BT の学習を追加した場合、loss の重みに関わらず全て訂正精度が減少していることもわかる。これらのことから、BT の学習が訂正モデルに悪影響を与えていると考えられる。一方で、Kiyono ら [26] は擬似データとして、対訳コーパスで学習した誤り生成モデルを用いて生成した擬似データを用いることで訂正モデルが向上することを報告している。彼らと本研究の違いとして、誤り生成モデルの性能差が考えられ、DAE のみで学習したモデルでは十分な誤りを生成できず、結果として BT の学習がうまくいっていないと考えられる。

5.4 統計的機械翻訳とニューラル機械翻訳の比較

表 5.3 に CoNLL-14 に対する USMT_{forward} iter 1 と、UNMT w/ DAE, UNMT w/ DAE, BT の出力を示す。一つ目の事例が前置詞に関するもので、gold は ‘in’ を削除している。二つ目の事例は主語に対応した動詞の誤りと、冠詞に関する誤りの 2 つを含んでいる。この場合、gold は ‘include’ を ‘includes’ にしており、定冠詞 ‘the’ を削除している。

一つ目の事例に対して USMT_{forward} iter 1 は正しく前置詞 ‘in’ を削除しており、他に余分な訂正も行っていない。UNMT w/ DAE も前置詞 ‘in’ は削除しているが、余分な訂正として前置詞 ‘up’ を挿入し、‘to ensure’ を削除している。UNMT w/ DAE, BT は前置詞 ‘in’ は削除せず、‘to ensure’ を削除している。このように、ニューラル手法を用いた場合は文意を変えている訂正を行っていることがわかる。

二つ目の事例に対して、USMT_{forward} iter 1 は正しく冠詞 ‘the’ を削除しているが、‘include’ を過去形に変化させている。この訂正は文法的には間違っていないが、入力文の文意を変えているため誤りとなる。UNMT の場合は BT の有無に関

表 5.3 CoNLL-14 に対する教師なし手法の出力例. 赤字は誤った箇所, 青字は訂正した箇所を示している.

source	Some will wish to keep it to themselves and hope to ensure that they will not bring in any pessimism into their family .
USMT _{forward} iter 1	Some will wish to keep it to themselves and hope to ensure that they will not bring any pessimism into their family .
UNMT w/ DAE	Some will wish to keep it up to themselves and hope that they will not bring any pessimism into their family .
UNMT w/ DAE, BT	Some will wish to keep it to themselves and hope that they will not bring in any pessimism into their family .
gold	Some will wish to keep it to themselves and hope to ensure that they will not bring any pessimism into their family .
source	The law ' s spirit also include the fairness .
USMT _{forward} iter 1	The law ' s spirit also included fairness .
UNMT w/ DAE	The law ' s spirit also include the fairness .
UNMT w/ DAE, BT	The law ' s spirit also include the fairness .
gold	The law ' s spirit also includes fairness .

ならず, 入力文をそのまま出力している. CoNLL-14 全体に含まれる冠詞の誤りの訂正率 (Recall) について調べたところ, USMT_{forward} iter 1 は 44.97 であった. 一方で UNMT w/ DAE は 17.06, UNMT w/ DAE, BT は 4.76 であり, USMT と比べると大きく差があることがわかった. これらのことから, ニューラル手法を用いた場合, 多くの冠詞の誤りは正しく訂正されていないと考えられる.

第6章 おわりに

近年、ニューラルネットワークを用いた研究が自然言語処理で盛んである。一般的にニューラルネットワークを用いた手法は訓練データとして大量のデータを必要とする。文法誤り訂正に関する研究でも学習者文とそれに対応した訂正文からなる大規模対訳コーパスを必要としている。一方で、学習者文と訂正文の組み合わせを用意するのは難しく、データ量が不足している問題に対していくつかの研究が行われている。

この問題に対して現在最も精度が良い手法として、単言語コーパスに対して擬似誤りを付与することで擬似対訳コーパスを作成し、このデータを用いてニューラルモデルも学習を行う手法が知られている。一方で本研究はこのデータ不足の問題に対して、機械翻訳で研究されている教師なし手法を用いた。この手法は訓練データとしてコンパラブルコーパスを必要とするので、機械翻訳システムを用いて作成した翻訳文を擬似学習者文として擬似コンパラブルコーパスを作成した。使用するデータ量を揃えた際には、この本研究の手法は擬似対訳データを使う場合と比べると、学習者の習熟度が高い入力に対しては高い訂正精度であることがわかった。しかし大量の擬似対訳データを用いたニューラル機械翻訳モデルに対しては、本研究の手法の訂正精度は遠く及ばない結果となった。また、ニューラル機械翻訳に基づく教師なし手法は機械翻訳の研究結果と異なり、文法誤り訂正では訂正精度は統計的機械翻訳に及ばなかった。これは逆翻訳を行う際の初期モデルの性能が低いためであると考えられる。

最適化手法が改善された際や擬似コンパラブルコーパスではなく、既存の学習者の記述した文を原言語側にした際には本研究の手法も改善すると考えられる。本研究の知見がラベルデータが少量である状況での文法誤り訂正の研究に役立ち、今後の学習者支援の研究全体の発展に対する一助となることを願っている。

謝辞

本論文の執筆に際して、小町守准教授には大変お世話になりました。研究室の先輩や後輩、同期にも多くのコメントを頂きました。大変感謝しております。また、Lang-8 のデータ使用に関して、株式会社 Lang-8 社長 喜洋洋氏に感謝いたします。

また小町守准教授や先輩の方々には研究というものについて多くのことを指導いただきました。特に山岸さん、松村さんには特に様々なことを教えていただき、深く感謝しております。研究のこと以外にも同期や先輩の皆様、後輩の方々には多くのことを教えていただきました。皆様のおかげでとても楽しい時間を過ごすことができました。

研究室に配属されてから3年間のうちで様々な方々と一緒に研究をすることができ、大変光栄でした。至らない点多々あったかと存じますが、多くの方と一緒に研究したことは大変勉強になりました。特に金子さんとは2年連続で一緒に後輩の研究を手伝う中で、多くのことを学ばせて頂きました。とても感謝しています。

最後に、副査を引き受けてくださった山口亨教授と高間康史教授に心より感謝いたします。

参考文献

- [1] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proc. of ACL*, pp. 789–798, 2018.
- [2] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Unsupervised statistical machine translation. In *Proc. of EMNLP*, pp. 3632–3642, 2018.
- [3] Mikel Artetxe, Gorka Labaka, and Eneko Agirre. An effective approach to unsupervised machine translation. In *Proc. of ACL*, pp. 194–203, 2019.
- [4] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *ICLR*, 2018.
- [5] Steven Bird. NLTK: The natural language toolkit. In *Proc. of COLING/ACL Interactive Presentation Sessions*, pp. 69–72, 2006.
- [6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.
- [7] Christopher Bryant and Ted Briscoe. Language model based grammatical error correction without annotated training data. In *Proc. of BEA*, pp. 247–253, 2018.
- [8] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In *Proc. of ACL*, pp. 793–805, 2017.
- [9] Christopher Bryant, Mariano Felice, Øistein E. Andersen, Ted Briscoe. The BEA-2019 shared task on grammatical error correction. In *Proc. of BEA*, pp. 52–75, 2019.
- [10] Christian Buck, Kenneth Heafield, and Bas van Ooyen. N-gram counts and language models from the common crawl. In *Proc. of LREC*, pp. 3579–3584, 2014.
- [11] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for

- measuring progress in statistical language modeling. In *Proc. of INTER-SPEECH*, pp. 2635–2639, 2014.
- [12] Shamil Chollampatt and Hwee Tou Ng. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proc. of AAAI*, pp. 5755–5762, 2018.
- [13] Alexis Conneau and Guillaume Lample. Cross-lingual language model pre-training. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pp. 7057–7067. Curran Associates, Inc., 2019.
- [14] Daniel Dahlmeier and Hwee Tou Ng. Better evaluation for grammatical error correction. In *Proc. of NAACL-HLT*, pp. 568–572, 2012.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, pp. 4171–4186, 2019.
- [16] Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. Can Markov models over minimal translation units help phrase-based SMT? In *Proc. of ACL*, pp. 399–405, 2013.
- [17] Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proc. of NAACL-HLT*, pp. 644–648, 2013.
- [18] Tao Ge, Furu Wei, and Ming Zhou. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*, 2018.
- [19] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *Proc. of ICML*, pp. 1243–1252, 2017.
- [20] Sylviane Granger. The computer learner corpus: A versatile new source of data for SLA research. In Sylviane Granger, editor, *Learner English on Computer*, pp. 3–18. Addison Wesley Longman, 1998.
- [21] Roman Grundkiewicz and Marcin Junczys-Dowmunt. Near human-level

- performance in grammatical error correction with hybrid machine translation. In *Proc. of NAACL-HLT*, pp. 284–290, 2018.
- [22] Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proc. of BEA*, pp. 252–263, 2019.
- [23] Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proc. of WMT*, pp. 187–197, 2011.
- [24] Marcin Junczys-Dowmunt and Roman Grundkiewicz. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proc. of EMNLP*, pp. 1546–1556, 2016.
- [25] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proc. of NAACL-HLT*, pp. 595–606, 2018.
- [26] Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. An empirical study of incorporating pseudo data into grammatical error correction. In *Proc. of EMNLP-IJCNLP*, pp. 1236–1242, 2019.
- [27] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL Demo Sessions*, pp. 177–180, 2007.
- [28] Guillaume Lample, Ludovic Denoyer, and Marc’Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *ICLR*, 2018.
- [29] Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *Proc. of EMNLP*, pp. 5039–5049, 2018.
- [30] Benjamin Marie and Atsushi Fujita. Unsupervised neural machine translation initialized by unsupervised statistical machine translation. *arXiv*

preprint arXiv:1810.12703, 2018.

- [31] Ning Miao, Hao Zhou, Lili Mou, Rui Yan, and Lei Li. CGMH: Constrained sentence generation by metropolis-hastings sampling. In *Proc. of AAAI*, 2019.
- [32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR Workshop*, 2013.
- [33] Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proc. of IJCNLP*, pp. 147–155, 2011.
- [34] Jakub Náplava and Milan Straka. Grammatical error correction in low-resource scenarios. In *Proc. of W-NUT*, pp. 346–356, 2019.
- [35] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In *Proc. of ACL-IJCNLP*, pp. 588–593, 2015.
- [36] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proc. of EACL*, pp. 229–234, 2017.
- [37] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 shared task on grammatical error correction. In *Proc. of CoNLL Shared Task*, pp. 1–14, 2014.
- [38] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proc. of ACL*, pp. 160–167, 2003.
- [39] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL Demo Sessions*, pp. 48–53, 2019.
- [40] Y. Albert Park and Roger Levy. Automated whole sentence grammar

- correction using a noisy channel model. In *Proc. of ACL*, pp. 934–944, 2011.
- [41] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proc. of ACL*, pp. 1715–1725, 2016.
- [42] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MASS: Masked sequence to sequence pre-training for language generation. In *Proc. of ICML*, pp. 5926–5936, 2019.
- [43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017.
- [44] Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proc. of NAACL-HLT*, pp. 619–628, 2018.
- [45] Zheng Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In *Proc. of NAACL-HLT*, pp. 380–386, 2016.
- [46] Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proc. of NAACL-HLT*, pp. 156–165, 2019.

発表リスト

筆頭論文

1. 勝又智, 松村雪桜, 山岸駿秀, 小町守. **ニューラル日英翻訳における RNN モデルと CNN モデルの出力分析**. NLP 若手の会第 12 回シンポジウム (YANS2017). September 3, 2017.
2. 勝又智, 松村雪桜, 山岸駿秀, 小町守. **ニューラル機械翻訳における共起情報を考慮した語彙選択**. 言語処理学会第 24 回年次大会 (NLP2018), pp.1058–1061. March 15, 2018.
3. Satoru Katsumata, Yukio Matsumura, Hayahide Yamagishi and Mamoru Komachi. **Graph-based Filtering of Out-of-Vocabulary Words for Encoder–Decoder Model**. In Proc. of ACL 2018 Student Research Workshop, pp.112–119. July 17, 2018.
4. 勝又智, 小町守, 真鍋章, 大頭威, 嶋崎優子. **node2vec を用いた障害レポートにおける故障原因推定**. 言語処理学会第 25 回年次大会 (NLP2019), pp.1045–1048. March 15, 2019.
5. 勝又智, 小町守. **教師なし文法誤り訂正**. 言語処理学会第 25 回年次大会 (NLP2019), pp.1391–1394. March 15, 2019.
6. Satoru Katsumata and Mamoru Komachi. **Towards Unsupervised Grammatical Error Correction using Statistical Machine Translation with Synthetic Comparable Corpus**. In arXiv e-prints, 1907.09724. July 24, 2019.
7. Satoru Katsumata and Mamoru Komachi. **(Almost) Unsupervised Grammatical Error Correction using Synthetic Comparable Corpus**. In Proc. of BEA, pp.134–138. August 2, 2019.
8. 勝又智, 小町守, 真鍋章, 谷本恒野. **障害レポートの分類問題に対するデータ選択を用いた BERT モデルの精度向上**. 言語処理学会第 26 回年次大会発表予定 (NLP2020). March 16–19, 2020.

共著論文

1. 甫立健悟, 松村雪桜, 勝又智, 金子正弘, 小町守. **敵対的生成ネットワークを用いた文法誤り訂正**. NLP 若手の会第 13 回シンポジウム (YANS2018). August 28, 2018.
2. Yukio Matsumura, Satoru Katsumata, Mamoru Komachi. **TMU Japanese–English Neural Machine Translation System using Generative Adversarial Network for WAT 2018**. In Proc. of WAT2018, pp.1112–1119. December 3, 2018.
3. 甫立健悟, 金子正弘, 勝又智, 小町守. **文法誤り訂正における単語編集率を用いた訂正度の制御**. 言語処理学会第 25 回年次大会 (NLP2019), pp.635–638. March 14, 2019.
4. Kengo Hotate, Masahiro Kaneko, Satoru Katsumata and Mamoru Komachi. **Controlling Grammatical Error Correction Using Word Edit Rate**. In Proc. of ACL 2019 Student Research Workshop, pp.149–154. July 30, 2019.
5. Masahiro Kaneko, Kengo Hotate, Satoru Katsumata and Mamoru Komachi. **TMU Transformer System Using BERT for Re-ranking at BEA 2019 Grammatical Error Correction on Restricted Track**. In Proc. BEA, pp.207–212. August 2 2019.
6. 高橋悠進, 勝又智, 小町守. **語形変化を考慮したロシア語の文法誤り訂正の検討**. NLP 若手の会第 14 回シンポジウム (YANS2019). August 27, 2019.
7. 山下郁海, 勝又智, 金子正弘, Imankulova Aizhan, 小町守. **英語からロシア語への転移学習を用いた文法誤り訂正**. NLP 若手の会第 14 回シンポジウム (YANS2019). August 27, 2019.
8. 白石奈緒美, 勝又智, 小町守. **Persona Graph を用いた地方議会会議録のキーワード抽出の検討**. NLP 若手の会第 14 回シンポジウム (YANS2019). August 27, 2019.
9. 平尾礼央, 新井美桜, 嶋中宏希, 勝又智, 小町守. **ニューラルネットワークを利用した日本語学習者の複数項目作文能力推定**. NLP 若手の会第 14 回シンポジウム (YANS2019). August 27, 2019.

10. 高橋悠進, 勝又智, 小町守. **学習者の誤り傾向を考慮した擬似データを用いた文法誤り訂正**. 言語処理学会第 26 回年次大会発表予定 (NLP2020). March 16–19, 2020.
11. 山下郁海, 勝又智, 金子正弘, Imankulova Aizhan, 小町守. **言語間での転移学習を用いたロシア語文法誤り訂正**. 言語処理学会第 26 回年次大会発表予定 (NLP2020). March 16–19, 2020.
12. 平尾礼央, 新井美桜, 嶋中宏希, 勝又智, 小町守. **複数項目の採点を行う日本語学習者の作文自動評価システム**. 言語処理学会第 26 回年次大会発表予定 (NLP2020). March 16–19, 2020.
13. 玉鴻飛, 黒澤道希, 勝又智, 小町守. **事前学習モデルを用いた中国語文法誤り訂正**. 言語処理学会第 26 回年次大会発表予定 (NLP2020). March 16–19, 2020.