文法誤り訂正のための自己改良戦略に基づくノイズ除去

三田 雅人 [1,2]　清野 舜 [1,2]　金子 正弘 [3,1]　鈴木 潤 [2,1]　乾 健太郎 [2,1]
[1] 理化学研究所　[2] 東北大学　[3] 首都大学東京
{masato.mita, shun.kiyono}@riken.jp, kaneko-masahiro@ed.tmu.ac.jp,
{jun.suzuki, inui}@ecei.tohoku.ac.jp

1 はじめに

文法誤り訂正（Grammatical Error Correction; GEC）は，入力文中に含まれる文法誤りを自動的に訂正するタスクである．GEC タスクは，構造的な類似性から文法誤りが含まれる文から誤りが含まれない文への機械翻訳 (Machine Translation; MT) タスクの一種と見做してモデル化されることが多い．これまでにも，GEC のための多くのニューラルエンコーダデコーダモデル（EncDec）が提案されており，顕著な結果を達成している [7].

MT では，EncDec の性能はデータ量の増加に伴い向上することが知られている [8]．一方で，GEC においては，データ量の増加が必ずしもモデルの性能向上に寄与しないことが報告されている．Lo ら [10] は EFCamDat[2] と呼ばれる約 200 万文対から構成される世界最大規模の学習者コーパスを初めて GEC に使用した．Lo らの報告によると，EFCamDat を用いて訓練した GEC モデルの性能が，約 72 万文対のより小さなデータセットで訓練したモデルよりも劣る結果となり，要因の一つに一貫性のない訂正の存在を指摘した．例えば，“discuss about” は常に “discuss” に訂正される必要があるが，このコーパス中には不適切に訂正されたものや訂正し忘れのようなものといった多くの誤りが含まれている（表 1）．このような訂正ミスによって生じた訂正における不整合はノイズとして機能し，モデルの訓練を妨げる可能性があると考えられる．

通常，学習者コーパスは，母語話者や訓練を受けた専門家らによって提供されるため，GEC コミュニティはこれまで人手で作られたこれらのデータについては訂正ミスの少ないクリーンなものであると想定する傾向があった．しかし，このようなデータセットを使用して GEC モデルの訓練を行う場合，データの品質により注意を払う必要ある．そこで本研究では，GEC における効果的なノイズ除去手法の設計を目的とする．

2 関連研究

1節でも述べた通り、GEC 分野ではこれまでモデルの訓練に使用するデータについて暗黙的にノイズの少ないものであると想定する傾向があった．そのため，既存データセットにおけるノイズについては未だに調査されておらず，我々の知る限り，GEC におけるノイズの除去に関する先行研究は存在しない．

その一方で，MT の分野ではデータ中のノイズは重要なトピックになりつつある．MT では，人手作成された対訳データの他に Web クロールを用いて自動獲得した対訳データも使用するため，文境界や文アラインメント，誤翻訳などの問題を伴うさまざまな品質の対訳データの存在が指摘されている．また，MT コミュニティはこの種の品質が翻訳の精度に潜在的に大きな影響を与えることについて以前にも増して認識してきている．例えば，Khayrallah ら [5] はノイズの種類とニューラル機械翻訳（NMT）への影響について調査した．このようなノイズに対処するための素直な解決策として，データフィルタリングアプローチがある．これは，ノイズを含むデータをフィルタリング（除外）し，高品質でより小規模な対訳データのみ保持するアプローチであり，これまで MT で多く行われてきた [4]. 実際，ノイズの多い対訳コーパスに対するフィルタリングタスクも近年開催されている [9].

(a) 不適切な訂正に起因するノイズ (False Positive)
訂正前: *I want to* ***discuss about** the education.*
訂正後: *I want to* ***discuss of** the education.*
(b) 訂正し忘れに起因するノイズ (False Negative)
訂正前: *We* ***discuss about** our sales target.*
訂正後: *We* ***discuss about** our sales target.*

表1: ノイズの例

3 ノイズ除去問題

3.1 文法誤り訂正におけるノイズ

本研究では，出力文（訂正文）に含まれる文法誤りをノイズと定義する．GEC では訓練対訳データの出力文は文法的に正しいという仮定を置いてモデルの訓練を行なっているため，理論上は，モデルは入力文に含まれる誤った単語およびフレーズから出力文中の訂正された正しい単語およびフレーズへの多対一の写像を学習する（例．**discuss* ***about/ of***→*discuss*）．しかし，不適切な訂正に起因するノイズ（表 1-（a））や訂正し忘れに起因するノイズ（表 1-（b））の存在により，以下の二つの観点でモデル訓練時のノイズとして機能する可能性があると考えられる．

- 不適切な訂正パターンを学習する可能性がある
- 教師あり学習の仕組み上，1 対多の写像の学習が行えないため学習効率が悪化する可能性がある

そのため，本研究ではこれらのノイズを訓練データから除去することを目的とする．なお，文法的な正しさは必ずしも一意に決まらない場合もあり，それゆえ訂正における不整合が生じる場合がある（例．*He undergoes periodic health checks* ****in**→**of/on** there own.*）．しかし，これらは GEC タスクに依存しない言語的曖昧性から生じる言語現

Algorithm 1: 自己改良戦略に基づく対訳データノイズ除去

Data: ノイズを含んだ対訳データ $\tilde{\mathcal{D}}$
Result: ノイズ除去済み対訳データ $\hat{\mathcal{D}}$
1 $\hat{\mathcal{D}} = \{\}$ // create empty set
2 $\tilde{\mathcal{D}}$ からベースモデル $\tilde{\theta}$ を訓練
3 **for** $(X, Y) \in \tilde{\mathcal{D}}$ **do**
4 $Y' = \text{Beam\_Search\_Decoding}(\text{Y}; \tilde{\theta})$
5 perplexity $PPL(Y)$ および $PPL(Y')$ を計算
6 **if** $PPL(Y) - PPL(Y') \geqq 0$ **then**
7 $\hat{Y} = Y'$
8 **else**
9 $\hat{Y} = Y$
10 $\hat{\mathcal{D}} = \hat{\mathcal{D}} \cup \{(X, \hat{Y})\}$
11 $\hat{\mathcal{D}}$ を用いて新たなノイズ除去済みモデル $\hat{\theta}$ を訓練

象であるため，本研究におけるノイズ除去の対象としない．

我々は，データ中のノイズは，先行研究 [10] により報告された EFCamDat に限らず GEC で扱うデータセットには必然的に含まれていると考える．その理由は，訓練コーパス構築過程の観点から説明できる．EFCamDat や Lang-8 [12] などのほとんどの学習者コーパスは，誤りを含む可能性のある学習者が書いた入力文と教師によって訂正された対応する出力文のログに基づいて構築されている．ここで出力文の作成方法に着目する．GEC では，入力文と出力文が同一言語であるため，誤った箇所のみを“編集”することで出力文が生成される．そのため，すべての訂正者の訂正精度が 100% でない限り，理論上は表 1のようなノイズは避けられないが，残念ながら人間の精度が 100% でないことは実験的に示されている [3]．したがって，コーパス生成過程の観点から，任意の GEC 訓練コーパスには，程度の差はあれ，一定のノイズが含まれていると考えられる．

3.2 ノイズ除去シナリオ

はじめに，GEC タスクを形式的に定義する．ここで，θ を GEC モデルにおける訓練可能なパラメータ集合，$\mathcal{D}$ を文法誤りが含まれる入力文 X と文法誤りが含まれない出力文 Y の文対からなる訓練データ，すなわち，$\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{n}$ とする．このとき，我々の目的は訓練データ $\mathcal{D}$ を用いたときの次の損失関数 $\mathcal{L}(\mathcal{D}, \theta)$ を最小化する最適なパラメータ集合 $\widehat{\theta}$ を獲得することである．

$$\mathcal{L}(\mathcal{D}, \theta) = -\frac{1}{|\mathcal{D}|} \sum_{(X,Y)\in\mathcal{D}} \log(p(Y|X, \theta)). \quad (1)$$

従来研究では，訓練データ $\mathcal{D}$ は“クリーン”な対訳データ $\hat{\mathcal{D}}$ であると仮定されていた．本研究で扱うノイズ除去シナリオでは，訓練データは“ノイズの含まれる”対訳データ $\tilde{\mathcal{D}}$ であることを仮定し，ノイズ除去を行うことで獲得されるノイズの少ない対訳データ $\hat{\mathcal{D}}$ を用いて新たなより良い GEC モデルを訓練する．

4 自己改良戦略に基づくノイズ除去

ノイズに対処するためのアプローチとして，MT でも一般的に用いられるフィルタリングアプローチがある．しかし，フィルタリングに基づくノイズ除去は次の理由により GEC には不適切である可能性が高い．

- GEC は MT と比べて利用可能なデータが限られた低リソースタスクである
- 訓練事例の大部分はモデルの訓練に部分的には役立つ場合がある

そのため，リソースの少ない GEC のような設定では，データを“捨てず”にデータ量を可能な限り減らさず有効活用できるようなノイズ除去手法が有効であると考えられる．

上記の動機に基づき，GEC のためのノイズ除去手法として自己改良戦略に基づくノイズ除去手法（self-refinement）を提案する．Algorithm 1に提案手法の概要を示す．提案手法の着想は，GEC モデルは比較的一貫した訂正を出力することが既定できるため，その性質を利用して訂正付きデータに対して自動的に再訂正することにある．具体的には，機械学習に基づく GEC モデル（ベースモデル）をノイズの含まれた対訳データ $\tilde{\mathcal{D}}$ で学習し，ベースモデルを用いてインスタンス全体で $\tilde{\mathcal{D}}$ における出力文を一貫させる．ノイズは訂正者の不注意または能力不足などのランダム要因によって引き起こされる．一方で、EncDec などの機械学習に基づく GEC モデルの場合，類似した文脈が与えられたときは常に一貫した予測を行う．ノイズを含んだ対訳データ $\tilde{\mathcal{D}} = \{(X_i, Y_i)\}_{i=1}^{n}$ が与えられとき，出力文 Y_i から新たな出力文 $\hat{Y}_i$ を生成する（4 行目）．モデルの予測の一貫性により，ノイズ除去済み対訳データ $\hat{\mathcal{D}} = \{(X_i, \hat{Y}_i)\}_{i=1}^{n}$ は結果的によりノイズの少ないデータになることが期待される．なお，提案手法はラベル付きデータを増やすことが目的ではないが，自己学習 (self-training) の一種と見做すこともできる．本アプローチの課題として，ベースモデルが一貫して不適切な訂正を引き起こしてしまう可能性があることである．例えば，表1の例において，ベースモデルが全て“discuss about”に訂正してしまう可能性がある．このような課題に対し，提案手法ではベースモデルが誤った予測を行なった場合に元の出力文を復元するようなフェールセーフ機構を導入している（5-9 行目）．このステップでは，ベースモデル出力 Y' を新たな出力文として採用するかどうかを判断するために，モデル出力 $PPL(Y')$ と元の出力文 $PPL(Y)$ それぞれの perplexity の計算し，比較を行う．

データセット	文数 (ペア)	分割	スコアラー
BEA-train	561,100	訓練	-
EFCamDAT	2,269,595	訓練	-
Lang-8 big	5,689,213	訓練	-
BEA-valid	2,377	開発	-
CoNLL-2014	1,312	評価	M^2 scorer & GLEU
JFLEG	1,951	評価	M^2 scorer & GLEU

表2: 実験に使用するデータセットの概要

5 実験

5.1 実験設定

データセット 実験で使用したデータセットは次の通りである．まず，GEC コミュニティで訓練データとして一般的

訓練データセット	CoNLL-2014				JFLEG			
	Prec.	Rec.	$F_{0.5}$	GLEU	Prec.	Rec.	$F_{0.5}$	GLEU
BEA-train (B)								
B w/o denoising (B-BL)	58.7	30.6	49.6	63.3	67.3	38.9	58.7	52.3
B w/ LM filtering (B-LM)	56.5	28.2	47.1	63.0	66.9	38.3	58.3	52.6
B w/ self-refinement (B-SR)	55.0	37.5	**50.3**	**64.2**	65.4	46.8	**60.5**	**54.8**
EFCamDat (E)								
E w/o denoising (E-BL)	48.5	24.0	40.3	61.3	69.3	38.0	59.5	53.7
E w/ LM filtering (E-LM)	48.5	25.8	41.2	61.7	68.5	39.6	59.7	54.2
E w/ self-refinement (E-SR)	55.2	32.4	**48.4**	**63.5**	69.4	48.5	**63.9**	**57.1**
Lang-8 big (L)								
L w/o denoising (L-BL)	59.8	41.3	54.9	65.9	74.5	51.6	68.4	58.1
L w/ LM filtering (L-LM)	60.6	42.8	55.9	66.3	74.3	52.4	**68.6**	59.1
L w/ self-refinement (L-SR)	58.1	51.1	**56.5**	**67.7**	71.3	59.8	**68.6**	**61.0**
B+E w/o denoising (BE-BL)	58.5	30.0	49.1	63.4	71.3	40.6	62.0	53.9
B+E w/ LM filtering (BE-LM)	59.1	27.9	48.3	63.3	71.6	41.0	62.3	54.7
B+E w/ self-refinement (BE-SR)	61.2	38.0	**54.5**	**65.2**	70.9	50.2	**65.5**	**58.0**
B+E+L w/o denoising (BEL-BL)	62.8	39.2	56.1	65.7	74.9	47.1	67.0	56.9
B+E+L w/ LM filtering (BEL-LM)	64.7	38.0	56.7	65.9	75.7	48.4	68.0	57.8
B+E+L w/ self-refinement (BEL-SR)	61.7	49.4	**58.8**	**68.0**	74.1	59.3	**70.6**	**61.4**

表3: 実験結果: **太字**の値は各訓練データセットにおける最高精度を示している．

に広く利用され，BEA-2019 Shared- Task の公式データセットとして採用された BEA，および先行研究 [10] でノイズの存在が指摘された EFCamDat を使用した．ここで，BEA-2019 Shared Task では訓練用（BEA-train）と開発用（BEA-valid）それぞれのデータセットが提供されているため，本実験においてもそれぞれ訓練/開発データセットとして使用した．さらに，EFCamDat 以外の大規模コーパスにおいてもノイズ除去の効果があるかを調査するために，Lang-8 big コーパス[*1]を訓練データとして使用した．各データセットの詳細は表 2に示す通りである．訓練データの前処理として，はじめに `spaCy tokenizer`[*2]を用いてトークナイズ処理を行った．次に，（1）無編集な文対，（2）入力文および出力文ともにトークン長が 80 以上の文対を訓練データから除外した．最後に，byte-pair-encoding（BPE）アルゴリズムを用いて，出力文からサブワードを取得した．サブワード化には，`subword-nmt`[*3]を使用し，マージ回数は 8,000 に設定した．

評価　GEC モデルの頑健な評価にはコーパス横断評価が必要であることが知られている [11]．そこで本評価実験においても，Mita ら [11] に従い，CoNLL-2014 評価データセット (CoNLL-2014) [15] および JFLEG 評価データセット (JFLEG) [13] の二つの評価データセットに対して，それぞれ M^2 scorer [1] および GLEU [14] の二つの評価尺度を用いてコーパス横断的に評価を行なった．なお，実験結果として報告するのは 3 つのランダムシードを用いて訓練した性能の平均値である．

モデル　GEC モデルの構築にあたり，seq2seq モデルのツールキット `fairseq` [16] における “Transformer（big）” を使用した．また，Optimizer には Adam [6] を使用し，学習率は Vaswani ら [18] と同じ設定である．フェールセーフ機構における言語モデルとして，GPT-2 [17] の Pytorch 実装[*4]を使用した．

比較手法　4節で述べた通り，我々はフィルタリングに基づくノイズ除去手法は GEC には適していないという仮説を持っている．この仮説を検証するために，本実験の比較手法として，ノイズ除去なし (w/o denoising) に加えて，言語モデルを用いたフィルタリングに基づくノイズ除去手法 (LM-filtering) を用意した．LM-filtering は，出力文が入力文よりも流暢でない場合，出力文にはノイズが含まれているという仮説に基づく手法である．具体的には，まずノイズの含まれる対訳データ $\tilde{\mathcal{D}}$ の文対それぞれに対し，perplexity を計算する．次に，入力文の perplexity が出力文の perplexity よりも低い文対を除外する．ここで，perplexity の算出するために使用する言語モデルは，提案手法のフェールセーフ機構と同様に GPT-2 を使用した．

5.2　結果

表 3に実験結果を示す．まず，議論を簡単にするために，CoNLL-2014 における $F_{0.5}$ に着目する．実験結果より，B-BL は E-BL よりも性能が大幅に高いことがわかる（B-BL = 49.6 vs E-BL = 40.3）．これは，先行研究 [10] の結果とも一致している．さらに，BEA と EFCamDat を単純に結合して訓練に使用した場合においても，性能が改善するどころかむしろ悪化させてしまうことが確認できる（B-BL = 49.6 vs BE-BL = 49.1）．これは，ノイズの存在により，データ量の増加が必ずしも GEC モデルの性能向上に寄与しないことを示唆する．

一方で，提案手法によってノイズ除去することで，BEA と EFCamDat を結合して訓練すると大幅に性能が改善されることがわかる（BE-BL = 49.1 vs BE-SR = 54.5）．次に，全ての評価データおよび評価尺度における結果に着目すると，Lang-8 big を含め全ての訓練データセットに対して提案手法を用いてノイズ除去することで性能が改善していることが確認できる．このことから，提案手法が任意の訓練データセットおよび評価データセットにおいて有効であることが示唆される．

[*1] Lang-8 big コーパスは，2012 年から 2019 年の期間で収集されたログを基に作成された世界最大規模の学習者コーパスである．

[*2] https://spacy.io/

[*3] https://github.com/rsennrich/subword-nmt

[*4] https://github.com/huggingface/transformers#PyTorch-models

Confusion set	E-BL (%)	E-SR (%)
*discuss about → *discuss about	66.7	49.5
*discuss about → discuss	33.0	50.2
*discuss about → *discuss in	0.3	0.3
*enter in → *enter in	61.6	31.7
*enter in → enter	38.4	68.3

表4: EFCamDat におけるノイズ除去前後の confusion set の例（訂正前 → 訂正後）.

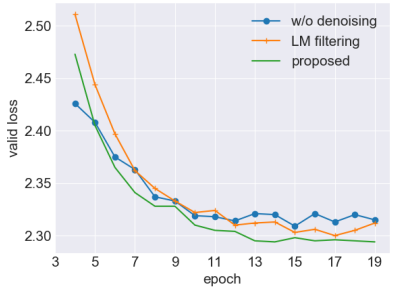

図1: EFCamDat を用いて訓練したときの validation loss

最後に，フィルタリングに基づくノイズ除去が GEC において有効であったかについて着目する．フィルタリングに基づくノイズ除去手法である LM-filtering は概ねノイズ除去をしない設定よりは性能が上回っているが，一方で，特に BEA のようにデータ量が少ない場合に適用するとかえって性能が悪化している（B-BL=49.6 vs B-LM=47.1）．これは我々の仮説の通り，たとえノイズを含む文対の除外に成功できたとしても，それと同時に本来 GEC モデルの訓練に部分的にでも有効であったインスタンスまでも除外してしまうからだと考えられる．

6 分析

本節では，提案手法によって実際にノイズが除去されたのか，またノイズ除去を行うことで，GEC モデルの学習効率は向上したのかについて分析する．ここで，学習効率が向上したとは，ノイズ除去前に比べてより良いパラメータ集合 $\widehat{\theta}$ を獲得したことを指している．

表 4はノイズ除去前後の EFCamDat データにおける confusion set を示している．冒頭でも例として挙げた “discuss about” に着目すると，ノイズ除去前の EFCamDat（E-BL）には “*discuss about→*discuss about” といった訂正し忘れに起因するノイズが全体の 66.7% 含まれているが，提案手法によってデータに対してノイズ除去を行うことで，49.5% まで削減することに成功し，結果として適切な訂正である “*discuss about→ discuss” という訂正も 33.0% から 50.2% まで改善していることがことが確認できた．他の事例として，“discuss” と同様に他動詞しかとらない動詞 “enter” を分析したところ，“*enter in→ *enter in” といったノイズの割合がノイズ除去前後で 61.6% から 31.7% まで大幅に削減できていることが確認できた．

ここまで，提案手法による訓練データ中のノイズ除去に一定の成果があったことを定量・定性の両面から確認してきたが，その結果として，GEC モデルの学習効率は向上したのだろうか．図1は EFCamDat を用いてモデルを学習したときの validation loss を示している．ここで，x 軸はエポックの推移，y 軸は validation loss をそれぞれ表している．この図から，提案手法（E-SR）の loss がノイズ除去なし（E-BL）やフィルタリングに基づいたノイズ除去（E-LM）の loss と比べて低い値になっていることがわかる．これは，モデルがノイズとなる訓練事例に惑わされなくなったことでより良いパラメータに落ち着きやすくなったと考えられる．

7 おわりに

本研究では，GEC の対訳データにおけるノイズ除去問題について初めて取り組んだ．また，GEC のための効果的なノイズ除去として，自己改良戦略に基づくノイズ除去手法（self-refinement）を提案した．実験結果より，訓練データセットにノイズ除去を行うことで性能が大幅に改善することを確認した．また，低リソースタスクである GEC において，提案手法は MT で一般的に行われるフィルタリングに基づくノイズ除去手法に比べて有効であることを確認した．なお，提案手法が有効に機能する境界条件については本研究で未調査なため今後の課題としたい．

参考文献

[1] Daniel Dahlmeier and Hwee Tou Ng. “Better Evaluation for Grammatical Error Correction”. In: *NAACL*. 2012, pp. 568–572.

[2] Jeroen Geertzen, Dora Alexopoulou, and Anna Korhonen. “Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge Open Language Database (EFCAMDAT)”. In: 2013.

[3] Roman Grundkiewicz and Marcin Junczys-Dowmunt. “Near Human-Level Performance in Grammatical Error Correction with Hybrid Machine Translation”. In: *NAACL*. 2018, pp. 284–290.

[4] Marcin Junczys-Dowmunt. “Dual Conditional Cross-Entropy Filtering of Noisy Parallel Corpora”. In: *WMT*. 2018, pp. 888–895.

[5] Huda Khayrallah and Philipp Koehn. “On the Impact of Various Types of Noise on Neural Machine Translation”. In: *WNMT*. 2018, pp. 74–83.

[6] Diederik Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *ICLR*. 2015.

[7] Shun Kiyono et al. “An Empirical Study of Incorporating Pseudo Data into Grammatical Error Correction”. In: *EMNLP-IJCNLP*. 2019, pp. 1236–1242.

[8] Philipp Koehn and Rebecca Knowles. “Six Challenges for Neural Machine Translation”. In: *Workshop on Neural Machine Translation*. 2017, pp. 28–39.

[9] Philipp Koehn et al. “Findings of the WMT 2018 Shared Task on Parallel Corpus Filtering”. In: *WMT*. 2018, pp. 726–739.

[10] Yu-Chun Lo et al. “Cool English: a Grammatical Error Correction System Based on Large Learner Corpora”. In: *COLING*. 2018, pp. 82–85.

[11] Masato Mita et al. “Cross-Corpora Evaluation and Analysis of Grammatical Error Correction Models — Is Single-Corpus Evaluation Enough?” In: *NAACL*. 2019, pp. 1309–1314.

[12] Tomoya Mizumoto et al. “Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners”. In: *IJCNLP*. 2011, pp. 147–155.

[13] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. “JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction”. In: *EACL*. 2017, pp. 229–234.

[14] Courtney Napoles et al. “GLEU Without Tuning”. In: *arXiv preprint arXiv:1605.02592* (2016).

[15] Hwee Tou Ng et al. “The CoNLL-2014 Shared Task on Grammatical Error Correction”. In: *CoNLL*. 2014, pp. 1–14.

[16] Myle Ott et al. “fairseq: A Fast, Extensible Toolkit for Sequence Modeling”. In: *NAACL*. 2019.

[17] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: (2019).

[18] Ashish Vaswani et al. “Attention Is All You Need”. In: *NIPS*. 2017, pp. 5998–6008.