Vol.2022-NL-253 No.18
2022/9/30

後続文脈の考慮が文法誤り訂正性能にもたらす影響の調査

井手 佑翼[1,a)]　出口 祥之[1]　五藤 巧[1]　Armin Sarhangzadeh[1]　渡辺 太郎[1]

概要：既存の典型的な文法誤り訂正モデルは各入力文を独立に扱うため，文脈を考慮した訂正を行えない．この問題に対して先行研究では，訂正対象の文だけでなく，先行する文脈をモデルに入力する手法が提案されてきた．本研究は，これに加えて後続の文脈または前後両方の文脈を入力した場合に訂正性能がどのように変化するか，定性分析を交えて調査する．

Studies of the Impact of Subsequent Context Information in Grammatical Error Correction

1. はじめに

文法誤り訂正（Grammatical Error Correction, GEC）は，与えられたテキストに含まれる誤りを訂正するタスクである．GEC モデルによる訂正の性能は，ニューラル機械翻訳手法の導入などにより大きく向上してきた．

しかし既存の典型的な GEC モデルは，テキストを文単位で訂正するという制約がある．すなわち各文を独立に処理するため，訂正の手がかりが文をまたいで存在するような誤りを正しく訂正できない．そうした誤りを正しく訂正するためには，訂正対象文に加えて文脈の情報をモデルに入力する必要がある．このアイデアに基づく手法は複数の先行研究で提案されてきたが，それらはいずれも訂正対象に先行する文のみを考慮するものであり，訂正対象の後ろにある手がかりを利用できない．たとえば以下の短い文書において第 1 文が訂正対象であるとき，*go* は訂正不要であるか，あるいは will go や went などに直すべきかを判断するためには，後続文の時制などを考慮する必要がある．

(a) Today I *go* to school. I attended a lecture there.

そこで本研究では，先行する文脈，後続の文脈，あるいは前後両方の文脈を入力した場合のそれぞれについて，GEC モデルの性能が改善するか調査する．モデルにはシンプルな単一 Encoder のモデルを用い，訂正対象の原文に文脈を加えた系列から訂正済みの一文を生成する．また，これに変更を加えたモデルを 2 つ提案する．1 つ目の Segment Embedding では，各トークンに，先行文脈，訂正対象文，後続文脈のいずれに属するかを表すベクトル表現を足し合わせる．2 つ目の Context-Aware Dropout では，訂正対象文より文脈が重視されるようドロップアウト率を調整する．

2. 関連研究

文脈を考慮した GEC の先行研究としては，Chollampatt ら [4] や Yuan ら [18] がある．これらはいずれも Sequence-to-Sequence（Seq2Seq）モデルを用いた研究であるが，このうち Yuan らは，単一 Encoder のモデルと，複数 Encoder で訂正対象文と文脈を別々にエンコードするモデルとを比較し，後者でより高い性能が得られたと報告している．Yuan らはさらに，この複数 Encoder モデルを用いて文脈文数と訂正性能の関係を調べ，評価データによる差はあるものの 1 文から 2 文を文脈として用いるのが最適であったと報告している．

機械翻訳においても，同じ Seq2Seq モデルで文脈の考慮に取り組んだ研究が多く存在する．その手法は，追加の Encoder を用いることで文単位の翻訳に文脈を与えるもの，そして翻訳する単位を一文から複数文に拡張するものに大別できる [1]．このうち後者に属する Zhang ら [19] は，標準的な Transformer ベースの単一 Encoder モデルで，より大規模な複数 Encoder/Decoder のモデルに匹敵する性能を得られたことを報告している．Fernandes ら [7] は文脈と翻訳対象文を結合した系列を Transformer に入力する手法を用いたが，彼らは加えて文脈情報が重要であるという

[1] 奈良先端科学技術大学院大学
[a)] ide.yusuke.ja6@is.naist.jp

バイアスをモデルに与えるため，CoWord Dropout を提案した．これは，翻訳対象文のトークンから一部をランダムに選び，mask に置き換えるものである．

3. 提案手法

3.1 前後の文脈を考慮した GEC

前後の文脈を考慮したモデル（文脈考慮モデル）では，Yuan ら [18] の単一 Encoder モデルと同様に，訂正対象文と文脈を結合した系列を入力する．入力に含める前後の文脈長は，実験設定ごとに変化させる．先行する n 文および後続の m 文を含める条件は，先 n 後 m と呼ぶことにする．訂正対象文と先行文脈，後続文脈を結合するための特殊なトークンとして，それぞれ <c1>，<c2> を用いるが，対応する文脈が 0 文の場合はこれを挿入しない．たとえば，1 節の例 (a) の第 1 文が訂正対象で，これを先 0 後 1 条件で処理する場合は，入力が

Today I *go* to school. <c2> I attended a lecture there.

出力される訂正文が

Today I *went* to school.

のようになる．ただし，訂正対象が文書の先頭や末尾に近く，指定された数の文を文脈として取れないケースでは，可能なかぎりで最も条件に近い文数を文脈として使用する．たとえば上の例を先 1 後 1 条件で処理する場合，訂正対象は最初の文であるため，先行文脈は 0 文と考え，既述の先 0 後 1 と同じ条件で処理を行う．以上は，次のように定式化される．

$$P_\theta(y^{(i)}|x^{(i)}) = \prod_{t=1}^{T} p_\theta(y_t^{(i)}|x_t^{(i)}, c_{precede}^{(i)}, c_{follow}^{(i)}, y_{<t}^{(i)}) \tag{1}$$

$$c_{precede}^{(i)} = x_{\geq max\{t-p,1\},\leq t-1}^{(i)}, c_{follow}^{(i)} = x_{\geq t+1,\leq min\{t+f,T\}}^{(i)} \tag{2}$$

なお，$x^{(i)}$，$y^{(i)}$ はそれぞれ原文側，訂正文側の第 i 文，T は $y^{(i)}$ の系列長，p は先行文脈長，f は後続文脈長，θ はパラメータを表す．

3.2 Segment Embedding

Segment Embedding ありモデルでは，入力系列の各トークンをベクトル空間に埋め込んだのち，それが先行文脈，訂正対象文，後続文脈のいずれに属するかに応じて，異なるベクトル表現を足し合わせる．これにより，3 種類の文を区別して入力する．

3.3 Context-Aware Dropout

Context-Aware Dropout（以下 CA Dropout）ありモデルでは，文脈情報の入力に加え，訂正対象文に含まれるトークンの埋め込み表現に対するドロップアウト率を上昇させる．つまり文脈をより重視した訂正がされるよう，訂正対象文の情報が相対的に多く隠された状態で学習を行う．ベースおよび訂正対象文のドロップアウト率をそれぞれ p_{base}，$p_{current}$ とすると，CA Dropout は次のように定式化される．

$$r_j^{(i)} \sim Bernoulli(1-p)\frac{1}{1-p} \tag{3}$$

$$\mathbf{y^{(i)}} = \mathbf{r^{(i)}} \odot \mathbf{x^{(i)}} \tag{4}$$

ただし，

$$p = \begin{cases} p_{current} & (i \in current) \\ p_{base} & (i \notin current) \end{cases} \tag{5}$$

であり，$\mathbf{x^{(i)}}, \mathbf{y^{(i)}}$ は，それぞれドロップアウト処理前，後のベクトル表現の系列，$current$ は訂正対象文に属するトークンのインデックスの集合，$\odot$ はアダマール積を表す．

本研究では，ベースとなる文脈考慮モデルのドロップアウト率を一律で 0.3 とするのに対し[*1]，CA Dropout ありモデルでは，訂正対象文に対してはドロップアウト率 $p_{current} = 0.4$ を用いる．

4. 実験

4.1 データセット

表 1 データセット概要

	データセット	使用文数	文数	文書数
訓練	FCE-train	17,824	29,145	2116
	W&I-train	22,550	35,090	3000
	NUCLE	20,049	58,038	1397
	Lang-8	1,074,910	2,032,319	177,847
	計	1,135,333	2,154,592	184,360
開発	FCE-dev	-	2,264	159
テスト	BEA-dev	-	4,384	350
	CoNLL-2014	-	1,312	50

本研究で使用するデータセットを**表 1** に示す．訓練データには First Certificate in English (FCE) corpus [17]，Write & Improve corpus [2]，National University of Singapore Corpus of Learner English (NUCLE) [6]，そして Lang-8 Learner Corpora v2.0 [10] を，開発データには FCE を用いる．これらのデータについては前処理として spaCy v3.3.0 `en_core_web_sm` による文分割およびトークン化を行った[*2]．また Lang-8 については，英語学習者によるエッセイから，少なくとも英文を 2 文，訂正アノテーションを 1 つ含むものを抽出して使用する．これは Chollampatt ら [4]

[*1] このドロップアウト率は，先 1 後 1 条件モデルを用いた予備実験において 0.1, 0.2, 0.3, 0.4 のうちで最も高いスコアを示したものである．

[*2] これらの処理を独自に行ったのは，公式に提供されているデータセットで使われた spaCy は v1.9.0 と古く，文分割の誤りが見られたことによる．

表 2 ハイパーパラメータ

アーキテクチャ	Transformer (base)
最適化手法	Adam ($\beta_1 = 0.9, \beta_2 = 0.98$)
損失関数	Label smoothed cross entropy ($\epsilon_{ls} = 0.1$)
更新回数	70,000
バッチサイズ	256
デバイス	1 NVIDIA RTX A6000
学習率	0.0005
学習率スケジューラ	Inverse square root
ドロップアウト率	0.3
クリッピングノルム	1.0
ビームサイズ	12

による前処理を踏襲した操作であり，この中で言語の特定には Lui ら [9] による langid.py を用いた．

加えて，以上を統合した訓練データ全体から，次の条件を満たす文対のみを抽出する．

- 原文と訂正文の間に違いがある．
- BPE（後述）適用後，その文と前後 3 文の文長（トークン数）がすべて 1 以上 80 以下である．

ここで抽出した文対の数は，表 1 に使用文数として示す．

以上のデータセットは BEA-2019 Shared Task [2] に準ずる内容だが，本研究では文脈情報を必要とする事情から，文書単位で公開されているデータセットしか使用できないという制約がある．そのためテストデータには BEA-2019 [2] のテストデータではなく開発データ（BEA-dev）を用いる．また，CoNLL-2014 [11] のテストデータも用いる．

4.2 実験設定

モデルには Transformer [16] を用いる．ハイパーパラメータは，[18] および Transformer (base) [16] を参考に**表 2** のとおりとした．ツールは，Fairseq [13] を使用した．サブワード分割には Byte Pair Encoding（BPE）[14] を採用し，学習は訓練データの訂正文側から 8000 回のマージ操作により行った．文脈長については，2 で見た [18] らの実験結果を踏まえ，前後の文脈長の合計が 3 文以下となるような組み合わせについて実験を行う．

4.3 評価・分析

評価には，既存の文単位・参照あり手法を用いる．BEA-dev の評価では ERRANT Scorer [3]，CoNLL-2014 では MaxMatch Scorer [5] により生成文から $F_{0.5}$ スコアを算出し，モデルおよび文脈条件間の比較を行う．また文脈なし条件を含む各文脈条件の間で，生成される文にどのような違いがあるか定性的に分析する．

5. 実験結果

5.1 評価スコア

モデル別，文脈条件ごとの評価スコアを**表 3** に示す．縦軸と横軸の数値は，それぞれ考慮する先行文脈と後続文脈の文数を表す．最上段の文脈なし&文脈考慮では，先 0 後 0 の欄に文脈なし条件の結果を示した．

まず BEA-dev のスコアに注目して結果を見ると，文脈考慮モデルでは，特に後続文脈を考慮する条件の多くで，文脈なし条件を上回るスコアが得られたことが分かる．一方，先行文脈のみを考慮した条件では，評価スコアが低下したものが複数見られた．考慮する文脈長が同じとき，先行文脈より後続文脈による性能改善が大きい傾向は，モデルによらず見られた．Segment Embedding ありモデルでは，ベースの文脈考慮モデルと比べて改善が見られない条件もあったが，先 1 後 1 条件では 34.93 の $F_{0.5}$ スコアが得られた．これは，全モデルを通じて最も高い BEA-dev のスコアである．CA Dropout ありモデルについては，文脈考慮モデルよりスコアが低下する条件が多かった．

一方，CoNLL-2014 のスコアからは，一部異なる結論が得られる．先行文脈と後続文脈それぞれを同じ文脈長で考慮した場合の比較では，先行文脈の方が高い性能を示した箇所が多く，BEA-dev とは逆の傾向が見られる結果となった．ただし，モデル間の比較で Segment Embedding ありモデルが最もよい性能を示す点は，BEA-dev による評価の結果と共通しており，先 3 後 0 条件で 52.78 の $F_{0.5}$ スコアが得られた．

5.2 定性分析

本節では，動詞の時制に関わる誤りに注目してケーススタディを行う．1 節で述べたように，動詞の時制の誤り訂正には文脈情報が必要であると考えられる．そこで，これを含む訂正の事例を観察し，文脈情報が訂正結果にもたらす影響を定性的に分析する．

動詞の時制に関する誤り訂正事例を，**表 4** に示す．原文および参照文は，BEA-dev データセットから抽出したものである．これに加え，文脈なし条件と先 1 後 2 条件の出力を挙げる．これらはいずれも，Transformer アーキテクチャをそのまま用いたモデルによる出力である．テキストに含まれる太字は訂正対象文，斜字は誤りまたは訂正箇所を表す．

この表から，文脈なし条件では *pronounce* を *pronounced* としており，受動態に訂正できていないのに対し，先 1 後 2 条件では *was pronounced* と時制・態の両方を正しく訂正できたことが分かる．この事例は先 2 後 1 条件でも正しく訂正されたが，先 1 後 1 や先 2 後 0 などの条件では *is pronounced* と態のみの訂正に留まったことから，計 3 文程度の文脈情報を入力することで初めて正しく訂正できるようになったと言える．その一方で *know* → *knew* など，文脈長を伸ばしても正しく訂正できない誤りが見られた．

表 3 モデル・文脈条件ごとの評価スコア

文脈なし&文脈考慮								
	BEA-dev				CoNLL-2014			
先＼後	0	1	2	3	0	1	2	3
0	32.68	32.08	32.82	33.70	50.85	49.20	48.42	51.31
1	33.14	32.98	**34.54**		**51.83**	51.01	50.64	
2	32.24	33.03			50.08	50.03		
3	31.87				50.62			
文脈考慮+Segment Embedding								
	BEA-dev				CoNLL-2014			
先＼後	0	1	2	3	0	1	2	3
0	-	32.45	34.87	33.69	-	48.64	50.35	50.66
1	32.39	**34.93**	32.67		50.38	50.85	49.92	
2	32.87	33.03			51.25	50.34		
3	33.28				**52.78**			
文脈考慮+CA Dropout								
	BEA-dev				CoNLL-2014			
先＼後	0	1	2	3	0	1	2	3
0	-	32.45	**33.92**	33.31	-	48.88	50.84	50.7
1	31.97	33.83	33.16		50.71	50.26	49.77	
2	32.42	33.00			50.79	49.94		
3	31.84				**52.10**			

表 4 文脈考慮モデルによる訂正事例

原文	There was a funny thing when I went to the marshal. The teacher call my name for twice and he told me to stand different place. **So I *know* that there was someone whose name *pronounce* same as me.** Near the race, I was very nervous. When I was running, I heard my housemates cheering and the wind passing my face.
参照文	So I *knew* that there was someone whose name *was pronounced* the same as mine .
文脈なし条件出力文	So I *know* that there was someone whose name *pronounced* the same as me .
先 1 後 2 条件出力文	So I *know* that there was someone whose name *was pronounced* the same as me .

6. おわりに

本研究は，文法誤り訂正モデルに対して，訂正対象文に加え，その前後の文脈を入力したときに訂正性能がどう変化するか調べた．提案手法の Segment Embedding および Context-aware Dropout の有効性を調査した結果, Segment Embedding ありモデルを用いた場合に最も大きなスコアの改善が見られた．ただし最適な文脈長は用いる評価スコアによって異なり，BEA-dev では前後 1 文ずつを入力する条件，CoNLL-14 では先行する 3 文のみを入力する条件で，最も高い性能が得られた．

今後の研究課題としては，今回用いた Seq2Seq と異なるモデルにおいて，文脈情報を入力する効果を調査することが挙げられる．具体的な例としては，系列タグ付けに基づくモデルを用いて文脈を考慮した GEC を行うことが考えられる．系列タグ付けは Omelianchuk ら [12] により文単位 GEC において高い性能を持つことが示されており，これに基づいた文脈考慮モデルによって，より高い性能を得られる可能性がある．

参考文献

[1] Bao, G., Zhang, Y., Teng, Z., Chen, B. and Luo, W.: G-transformer for document-level machine translation, *arXiv preprint arXiv:2105.14761* (2021).
[2] Bryant, C., Felice, M., Andersen, Ø. E. and Briscoe, T.: The BEA-2019 shared task on grammatical error correction, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 52–75 (2019).
[3] Bryant, C., Felice, M. and Briscoe, E.: Automatic annotation and evaluation of error types for grammatical error correction, Association for Computational Linguistics (2017).
[4] Chollampatt, S., Wang, W. and Ng, H. T.: Cross-sentence grammatical error correction, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 435–445 (2019).
[5] Dahlmeier, D. and Ng, H. T.: Better evaluation for grammatical error correction, *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 568–572 (2012).
[6] Dahlmeier, D., Ng, H. T. and Wu, S. M.: Building a large annotated corpus of learner English: The NUS corpus of learner English, *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pp. 22–31 (2013).

[7] Fernandes, P., Yin, K., Neubig, G. and Martins, A. F.: Measuring and increasing context usage in context-aware machine translation, *arXiv preprint arXiv:2105.03482* (2021).
[8] Kaneko, M., Mita, M., Kiyono, S., Suzuki, J. and Inui, K.: Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction, *arXiv preprint arXiv:2005.00987* (2020).
[9] Lui, M. and Baldwin, T.: langid. py: An off-the-shelf language identification tool, *Proceedings of the ACL 2012 system demonstrations*, pp. 25–30 (2012).
[10] Mizumoto, T., Komachi, M., Nagata, M. and Matsumoto, Y.: Mining revision log of language learning SNS for automated Japanese error correction of second language learners, *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 147–155 (2011).
[11] Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H. and Bryant, C.: The CoNLL-2014 shared task on grammatical error correction, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pp. 1–14 (2014).
[12] Omelianchuk, K., Atrasevych, V., Chernodub, A. and Skurzhanskyi, O.: GECToR–grammatical error correction: tag, not rewrite, *arXiv preprint arXiv:2005.12592* (2020).
[13] Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D. and Auli, M.: fairseq: A fast, extensible toolkit for sequence modeling, *arXiv preprint arXiv:1904.01038* (2019).
[14] Sennrich, R., Haddow, B. and Birch, A.: Neural machine translation of rare words with subword units, *arXiv preprint arXiv:1508.07909* (2015).
[15] Tiedemann, J. and Scherrer, Y.: Neural machine translation with extended context, *arXiv preprint arXiv:1708.05943* (2017).
[16] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Advances in neural information processing systems*, Vol. 30 (2017).
[17] Yannakoudakis, H., Briscoe, T. and Medlock, B.: A new dataset and method for automatically grading ESOL texts, *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 180–189 (2011).
[18] Yuan, Z. and Bryant, C.: Document-level grammatical error correction, *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, Online, Association for Computational Linguistics, pp. 75–84 (2021).
[19] Zhang, P., Chen, B., Ge, N. and Fan, K.: Long-short term masking transformer: A simple but effective baseline for document-level neural machine translation, *arXiv preprint arXiv:2009.09127* (2020).