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

甫立 健悟　金子 正弘　勝又 智　小町 守
首都大学東京
{hotate-kengo, kaneko-masahiro, katsumata-satoru}@ed.tmu.ac.jp,
komachi@tmu.ac.jp

1　はじめに

文法誤り訂正タスクでは，近年，ニューラルネットワークを用いた文法誤り訂正モデルの研究が盛んに行われており，それらのモデルは従来の統計的機械翻訳を用いた文法誤り訂正モデルよりも高い性能を示している [1]．

第二言語学習者が書いた文法的に誤った文をある評価者が文法的に正しい文へと訂正するとき，その評価者によりどの程度訂正を行うのかは異なる．例えば，文法誤り訂正タスクにおいて学習データとして一般的に使用される Lang-8 [6]，評価データとして使用される CoNLL-2014 [9] や JFLEG [8] では，1 文中における訂正の量という意味での**訂正度**が異なることが知られている [8]．また，同様に学習者によってどの程度訂正を求めているのかも異なる．しかし，既存の文法誤り訂正モデルは学習した単一の訂正度でのみ訂正を行っており，それらの異なる訂正度で訂正を行う手法の研究は行われていない．

そこで，我々は訂正度を制御可能なニューラル文法誤り訂正モデルを提案する．文法誤りが訂正されているデータ内において，1 文ごとの訂正度の情報を特殊トークンとして文に付与し，新たな学習データを作成する．ここで，訂正度を表す指標として**単語編集率**を用いる．単語編集率とは文中の単語がどれだけ書き換えられたのかを表す指標であるため，文法的誤りを含んだ文と，その誤りを訂正した文の単語編集率は，文の訂正度を表していると言える．CoNLL-2014 と JFLEG では，JFLEG の方が訂正度が大きいことが知られており，実際に図 1 に示すグラフからも，CoNLL-2014 よりも JFLEG の方が単語編集率が大きいため，単語編集率が訂正度を示していることがわかる．そして，新たに作成した学習データを用いてニューラルネットワークモデルの学習を行い，推論時に入力文の文に任意の訂正度の特殊トークンを付与することで，付与した単語編集率に基づきモデルの訂正度を制御する．

実験では，訂正度の異なる CoNLL-2014 と JFLEG

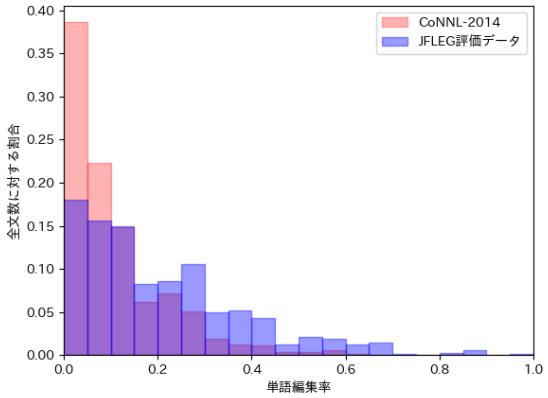

図 1: 1 文中の単語編集率のヒストグラム

でそれぞれスコアを求めることにより，モデルの訂正度を制御した上での訂正の質を調べた．その結果，学習時に付与した訂正度の情報により，実際にモデルの訂正度が制御できることを明らかにした．また，訂正度の情報を用いていないモデルと比較すると，付与した訂正度の情報と評価データの訂正度がマッチした場合，GLEU と $F_{0.5}$ の両方のスコアが向上することを示した．

2　先行研究

Junczys-Dowmunt ら [4] は統計的機械翻訳を用い，タスクに合わせたパラメータの調整を行うことにより，それまでの文法誤り訂正モデルの中で最高性能を出した．しかし，統計的機械翻訳を用いたモデルでは，単語やフレーズ単位での局所的な訂正しかできない．そこで，より文脈など，単語間の意味を考慮した流暢な訂正を行うためにニューラルネットワークを用いた手法が幾つか提案された．その中で，Chollampatt and Ng [1] は，畳み込みニューラルネットワーク（CNN）のエンコーダデコーダモデルを用いた手法を提案した．このモデルは，現在実装が公開されている [1] 文法誤り

[1] https://github.com/nusnlp/mlconvgec2018

表 1: 各学習データの概要

データ	文数
Lang-8	1.3M
NUCLE	50K
擬似データ（NYT 2007）	0.5M
Lang-8 + NUCLE + 擬似データ	1.7M [2]

訂正モデルの中で精度が高いモデルの一つであり，従来の統計的機械翻訳を用いたモデルよりも高い性能を示している．だが，これらのモデルでは，学習した訂正度でのみ訂正を行い，その訂正度を状況に応じて制御することはできない．

Kikuchi ら [5] は文書要約タスクにおいてエンコーダデコーダモデルに出力長を制限する情報を与えることで，出力長の制御を可能にした．また，Sennrich ら [11] は機械翻訳において学習データに文の丁寧さの情報を特殊トークンとして入力文に付与することで，出力文の敬意表現の制御を行った．Sennrich らと同様に，本研究では，訂正度を表す単語編集率を学習データに特殊トークンとして付与することで，入力文に対する訂正度の制御を行う．

3 単語編集率による訂正度制御

学習データ内の全文に対して単語編集率を求め，その値に基づいた特殊トークンを付与することで，モデルの訂正度を制御する手法を提案する．以下に単語編集率の求め方と，特殊トークンの付与の仕方について説明する．

はじめに，学習データ内の誤り文とそれに対応する訂正文から挿入の回数，削除の回数，置換の回数の和が最小となるように動的計画法を用いて編集距離を計算する．そして，求めた編集距離を誤り文の文長で割り，単語編集率を計算する．

算出した単語編集率をもとに学習データをソートし，文数が均等になるように幾つかの文集合に分割する．その分割した文集合ごとに異なる特殊トークンを定め，誤り文の文頭に付与する．

このようにして，文頭に単語編集率によって定められた特殊トークンが付与された誤り文とそれに対応する訂正文の学習データを作成した．この新たに作成した学習データを用いて文法誤り訂正モデルを学習する．

[2]学習データとしては前処理として訂正がある文のみを使用し，文長制限 80 をかけている．

表 2: 全学習データの中で特殊トークンとして区切られた単語編集率の閾値と文数

特殊トークン	最小値	最大値	文数
⟨1⟩	0.01	0.12	350K
⟨2⟩	0.12	0.20	350K
⟨3⟩	0.20	0.31	350K
⟨4⟩	0.31	0.53	350K
⟨5⟩	0.53	38.00 [3]	350K

4 訂正度制御実験

4.1 データ

表 1 に各学習データの概要を示す．本研究では，文法誤り訂正データとして Lang-8 と NUCLE [3] を用いた．評価データとしては，CoNLL-2014 評価データと JFLEG 評価データを用い，それぞれの開発データとして CoNLL-2013 [10] 開発データ と JFLEG 開発データを用いた．後述する擬似データ作成の際，誤りを付与する単言語データとしては，The New York Times Annotated Corpus（LDC2008T19）[4] の 2007 年のデータ（NYT 2007）のみを用いた．

4.2 擬似データ作成

文法誤り訂正の分野において，擬似データを学習データに追加することで精度が向上することが知られている [13] ．そこで，本研究では，学習データ量の増加のため，擬似データを作成し学習データに加えた上で文法誤り訂正モデルの実験を行った．擬似データ作成のための文法誤り生成モデルとしては Vaswani ら [12] の Transformer を用い [5]，ハイパーパラメータも全て同様の数値に設定した．学習データとしては，Lang-8 と NUCLE を合わせたデータの訂正文を原文，誤り文を正解文としたデータを用いた．

学習した文法誤り生成モデルに NYT 2007 を入力文として与え，誤りが付与された出力文と組み合わせることで擬似データを作成した．

4.3 文法誤り訂正モデル

本研究では，文法誤り訂正モデルとして CNN を用いた Chollampatt and Ng のモデル [1] を用い，ハイパーパラメータも同様の値を用いた．学習データとして，Lang-8 と NUCLE に擬似データを加えたデータ

[3]原文の単語数を超える単語数の挿入，削除，置換が行われる場合，単語編集率は 1 を超える．

[4]https://catalog.ldc.upenn.edu/LDC2008T19

[5]CNN モデルでも同様に擬似データ作成を試みたが ，有効な結果は得られなかった．

表 3: 訂正度を制御した文法誤り訂正結果（* はベースラインに対し統計的有意差があることを示す（$p < 0.05$））

モデル	CoNLL-2013（開発）			CoNLL-2014（評価）			JFLEG（開発）	JFLEG（評価）	
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	GLEU		WER
ベースライン	42.19	15.28	31.20	53.20	25.18	43.52	47.92	51.77	0.10
特殊トークン									
⟨1⟩	**52.45**	13.60	33.39	**60.07**	23.52	*45.83	44.85	*48.45	0.06
⟨2⟩	47.55	17.94	**35.75**	54.64	28.41	***46.12**	47.96	*52.01	0.09
⟨3⟩	43.38	20.05	35.19	50.48	31.45	*45.03	**49.45**	***53.59**	0.12
⟨4⟩	40.91	**21.32**	34.56	47.43	**32.68**	43.50	49.16	*53.47	0.17
⟨5⟩	29.48	13.98	24.13	33.77	22.95	*30.86	37.52	*42.21	0.43

を用いたモデル （ベースライン），合成データに単語編集率で定められた特殊トークンを付与したデータを用いたモデルの 2 種類の実験を行った．学習データに付与した特殊トークンとしては，単語編集率の大きさに準じて，表 2 に示すように ⟨1⟩（最も単語編集率が小さい文集合に付与）, ⟨2⟩, ⟨3⟩, ⟨4⟩, ⟨5⟩（最も単語編集率が大きい文集合に付与）の 5 種類の特殊トークンを用いた．

評価手法としては，CoNLL-2013 と CoNLL-2014 に対して適合率，再現率，$F_{0.5}$ スコア [2] で評価し，JFLEG に対しては GLEU [7] で評価した．また，JFLEG の評価データの入力文とそれに対するモデルの出力文から単語編集率（WER）を 1 文ずつ算出し，その値を全文数で割った平均値を算出した．

4.4 実験結果・考察

実験結果を表 3 に示す．表中の「特殊トークン」モデルについては，全て同一のモデルであり，推論時の入力文に対して付与した特殊トークンごとにスコアを出したものである．

表中の単語編集率（WER）から，実際の単語編集率の平均値が，入力文に付与された特殊トークンに比例して変化していることがわかる．つまり，単語編集率により定められた特殊トークンによって，実際のモデルの単語編集率の制御ができていることがわかる．

CoNLL-2013 の開発データで最も高い $F_{0.5}$ スコアを出したモデルは，特殊トークン ⟨2⟩ を付与して入力したときであり，その特殊トークンは CoNLL-2014 の評価データにおいても最も高い $F_{0.5}$ スコアを出している．同様に，JFLEG の開発データで最も高い GLEU スコアを出したモデルは，特殊トークン ⟨3⟩ を付与して入力したときであり，その特殊トークンは JFLEG 評価データにおいても最も高い GLEU スコアを出している．また，CoNLL-2014 と JFLEG では訂正度が異なる [8] ため，それらの訂正度にマッチした特殊トークンが異なり，最も高いスコアを出した特殊トークンが異なったと考えられる．言い換えると，単語編集率が特殊トークンで制御できたことにより，どちらの評価データでもベースラインより高いスコアを出すことができたと考えられる．

それぞれの特殊トークンでの適合率と再現率を見ると，特殊トークン ⟨1⟩ での適合率が最も高く，再現率が低い，一方で，特殊トークン ⟨4⟩ での適合率が最も低く，再現率が高くなっている．このことから，単語編集率と比例して再現率が変化し，反対に単語編集率と反比例して適合率が変化していることがわかる．

ただし，特殊トークン ⟨5⟩ の単語編集率は最も高いが，再現率は適合率と共に低い値となっている．また，特殊トークン ⟨5⟩ のスコアは，GLEU と $F_{0.5}$ の両方で低いスコアとなっている．この特殊トークンが付与されている学習データは，表 2 より，単語編集率が 0.53 以上のデータである．これらの学習データを確認したところ，文長が短い，もしくは，訂正とは関係のないコメントが挿入されているなどのノイズが大きいデータとなっていた．そのため，学習が上手く行かずスコアが上がらなかったと考えられる．

4.5 出力例

モデルの入力文に付与された異なる特殊トークンによる出力例を表 4 に示す．この表は，JFLEG の評価データに対する出力例であり，入力文に付与した特殊トークンごとにそれぞれ示している．太字になっている単語は，原文から変更がされた箇所を表している．

この例では，原文に対し，訂正すべき箇所が複数箇所ある．特殊トークン ⟨3⟩ ではそのうちの 2 箇所のみ訂正を行っているが，特殊トークン ⟨4⟩ で，訂正すべき箇所を全て網羅している．特殊トークン ⟨5⟩ では，誤っているが更に異なった変更を加えている．このことか

表 4: JFLEG の評価データに対する出力例

原文	Disadvantage is parking their car is very difficult .	WER
正解文	**The disadvantage** is **that** parking their car is very difficult .	0.33
ベースライン	Disadvantage is parking their car is very difficult .	0.00
特殊トークン		
⟨1⟩	Disadvantage is parking **;** their car is very difficult .	0.11
⟨2⟩	**Disadvantages are** parking their car is very difficult .	0.22
⟨3⟩	**The disadvantage** is parking their car is very difficult .	0.22
⟨4⟩	**The disadvantage** is **that** parking their car is very difficult .	0.33
⟨5⟩	**The disadvantage** is **that** their car **parking lot** is very difficult .	0.56

ら，実際に提案モデルが訂正度の異なる訂正を行っていることが確認できる．また，ベースラインの出力では訂正されていないが，提案モデルでは特殊トークンによって多くの訂正を行わせることで，全ての訂正すべき箇所を訂正することができている．

5 おわりに

本研究では，単語編集率に基づいた特殊トークンを付与した学習データを作成し，ニューラルネットワークを用いた文法誤り訂正モデルを学習することで，モデルの訂正度の制御が可能であることを示した．また，制御した訂正度が評価データとマッチした場合，GLEU と $F_{0.5}$ の両方でスコアが向上することも示した．

今後としては，単語編集率では得られない，単語単位の訂正であるのか，フレーズ単位での訂正であるのかという様なより詳細な情報を入れ，特殊トークンを付与することでより細かい粒度で制御する実験などに取り組みたい．

参考文献

[1] Shamil Chollampatt and Hwee Tou Ng. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proc. of AAAI*, 2018.

[2] Daniel Dahlmeier and Hwee Tou Ng. Better evaluation for grammatical error correction. In *Proc. of NAACL-HLT*, 2012.

[3] Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proc. of BEA*, 2013.

[4] Marcin Junczys-Dowmunt and Roman Grundkiewicz. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proc. of EMNLP*, 2016.

[5] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In *Proc. of EMNLP*, 2016.

[6] Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. The effect of learner corpus size in grammatical error correction of ESL writings. In *Proc. of COLING*, 2012.

[7] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In *Proc. of ACL-IJCNLP*, 2015.

[8] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. JFLEG: A fluency corpus and benchmark for grammatical error correction. In *Proc. of EACL*, 2017.

[9] 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*, 2014.

[10] Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. The CoNLL-2013 shared task on grammatical error correction. In *Proc. of CoNLL*, 2013.

[11] Rico Sennrich, Barry Haddow, and Alexandra Birch. Controlling politeness in neural machine translation via side constraints. In *Proc. of NAACL-HLT*, 2016.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NIPS*. 2017.

[13] 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*, 2018.