文法誤り訂正における複数の逆翻訳モデルを利用した訂正傾向の比較

小山 碧海　甫立 健悟　金子 正弘　小町 守
東京都立大学
{koyama-aomi, hotate-kengo, kaneko-masahiro}@ed.tmu.ac.jp
komachi@tmu.ac.jp

1 はじめに

文法誤り訂正（GEC）とは，言語学習者の書いた文にある誤りを自動で訂正するタスクである．Yuan ら [1] がエンコーダ・デコーダ（EncDec）モデルを GEC に適用して以来，様々な EncDec ベースの GEC モデルが提案されている [2, 3, 4]．また，GEC モデルは構造ごとに異なる訂正傾向を持つことが知られている．例えば，CNN [5] ベースのモデルは，局所的な誤りの訂正性能が高いことが示されている [3]．さらに，複数の GEC モデルを組み合わせることにより，性能を向上させることも行われている [6, 7]．一方で，EncDec モデルの訓練には大量の訓練データが必要であり [8]，GEC では訓練に使用可能な学習者データが不足しているという問題がある．そのため，様々な擬似データ生成手法が研究されており [4, 9, 10]，大量の擬似データを利用した EncDec モデルが顕著な性能を達成している [11, 12, 13]．

代表的な擬似データ生成手法の一つに逆翻訳 [14] がある．逆翻訳では，GEC モデルの場合とは反対に，訂正文から学習者文を出力させるように逆翻訳モデルを訓練する．その後，逆翻訳モデルに対して文法的に正しい文を入力し，擬似的な誤りを含む文を生成する．このようにして得た擬似誤り文とその入力文のペアを擬似データとして GEC モデルの訓練に使用する．Kiyono ら [12] は，いくつかの擬似データ生成手法を比較した結果，逆翻訳による擬似データで事前学習された GEC モデルの性能が最も高かったと報告しており，逆翻訳は最も効果的な擬似データ生成手法の一つであると考えられる．

我々は，GEC モデルの場合と同様に，逆翻訳モデルの構造によって GEC モデルの訂正傾向が異なる可能性があると考えた．しかしながら，これまでの研究では逆翻訳モデルに GEC モデルと同じモデルを使用することが多く [9, 12, 15]，異なる逆翻訳モデルを使用した場合にどのような訂正傾向の違いがあるかを比較していない．また，Wan ら [16] は潜在表現にノイズを加えることによる擬似データ生成手法とルールベースの擬似データ生成手法を比較し，擬似データごとに訂正傾向が異なることを報告している．さらに，それらの擬似データを組み合わせることにより GEC モデルの性能を向上させている．したがって，訂正傾向の違いを捉えることは，より高性能な GEC モデルを構築することに繋がる．

そこで，我々は 3 つの EncDec モデル（Transformer [17]，CNN，LSTM [18]）を逆翻訳モデルとして使用し，それらが生成した擬似データで事前学習された GEC モデルの訂正傾向を調査した．その結果，逆翻訳モデルごとに誤りタイプ別の訂正傾向が異なることが明らかになった．さらに，異なる逆翻訳モデルから生成した擬似データを組み合わせた場合の訂正傾向を調査した．その結果，異なる逆翻訳モデルから生成した擬似データを組み合わせた場合，シードのみが異なる単一の逆翻訳モデルを使用した場合に対して，性能が向上するあるいは補間する性能となることを確認した．

2 関連研究

Htut ら [19] は複数の GEC モデル（Transformer，CNN，PRPN [20]，ON-LSTM [21]）に対して，異なる逆翻訳モデル（Transformer，CNN）から生成された擬似データを使用した場合の訂正性能を調査した．その結果，GEC モデルに Transformer を使用し，逆翻訳モデルに CNN を使用した場合に最も高い性能を達成したと報告した．ただし，GEC では擬似データを事前学習に使用することが一般的であるが [4, 10, 12]，Htut ら [19] は擬似データを学習者データで訓練した後の再訓練に使用するというあま

り一般的ではない方法を用いている．さらに，Htut ら [19] は擬似データごとにどのような訂正傾向があるかについて報告していない．我々は，GEC モデルに Transformer を使用し，異なる逆翻訳モデル（Transformer，CNN，LSTM）を使用した場合の訂正傾向を調査する．また，Kiyono ら [12] に従い，擬似データを GEC モデルの事前学習に使用する．

White ら [22] は，直接的な編集操作による擬似データ生成手法 [11, 23] の比較を行なった．1 つ目の手法 [11] では，スペルチェッカーに基づいて構築された confusion set を利用して擬似データを生成する．2 つ目の手法 [23] では，学習者データから抽出された誤りパターン，および動詞・名詞・前置詞の置換を利用して擬似データを生成する．2 つの手法 [11, 23] を比較した結果，1 つ目の手法 [11] ではスペリングの誤りの訂正性能が高いこと，および 2 つ目の手法 [23] では名詞の単数・複数形と時制の誤りの訂正性能が高いことを報告している．我々は，異なる逆翻訳モデルから生成された擬似データを用いた場合の GEC モデルの訂正傾向を報告する．

いくつかの研究 [10, 16, 24] では，異なる手法により生成された擬似データを組み合わせて GEC モデルの訓練に使用している．例えば，Zhou ら [24] は，統計的機械翻訳からの翻訳文を擬似誤り文とし，ニューラル機械翻訳からの翻訳文を擬似訂正文とする擬似データ生成手法を提案し，さらに逆翻訳により生成された擬似データを組み合わせて GEC モデルの訓練に使用している．しかしながら，Zhou ら [24] は，擬似データを組み合わせた場合の GEC モデルの訂正傾向について報告していない．我々は，異なる逆翻訳モデルから生成した擬似データを組み合わせた場合の GEC モデルの訂正傾向を報告する．

3 実験設定

3.1 データセット

訓練データおよび検証データには BEA-2019 [25] で使用されたものを用いた．BEA-2019 で使用されたデータセットは FCE [26]，Lang-8 コーパス [27, 28]，NUCLE [29]，および W&I+LOCNESS [30, 31] から構成されている．Chollampatt ら [3] に従い，訓練データから編集のない文対を削除した．その後，訓練データの訂正文側のみからサブワードを獲得し，BPE [32] を適用した．ここで，語彙サイズは 8,000 とした．以降では，訓練データおよび検証データをそれぞれ BEA-train，BEA-valid と呼ぶ．

また，擬似誤り文を生成するための生成元コーパスとして，Wikipedia からランダムに抽出した 900 万文を使用した[1)]．

3.2 性能評価

GEC モデルの評価データには，CoNLL-2014 [33]，JFLEG [34]，および BEA-2019 のテストデータ（BEA-test）を使用した．CoNLL-2014 では M^2 [35]，JFLEG では GLEU [36] を評価指標に用いた．また，BEA-test および BEA-valid は，ERRANT [37, 38] を用いて評価を行なった．アンサンブルモデルを除く報告する全ての値は 3 つの異なるシードを用いて訓練された GEC モデルのスコアの平均である[2)]．アンサンブルモデルでは，平均値を得るために訓練した 3 つの GEC モデルをアンサンブルした結果を報告する．

3.3 文法誤り訂正モデル

GEC モデルには，代表的な EncDec ベースのモデルである Transformer を使用した．モデルのアーキテクチャは Vaswani ら [17] の "Transformer (base)" とし，fairseq [39] にある実装を用いた．Kiyono ら [12] に従い，擬似データを事前学習に使用し，その後 BEA-train でファインチューニングを行なった．事前学習では Adam [40]，ファインチューニングでは Adafactor [41] を最適化に用いた．さらに，ベースラインとして，事前学習を行わずに BEA-train のみを用いて訓練したモデルを用意した．

また，異なる逆翻訳モデルから生成した擬似データを組み合わせた場合の訂正傾向を調査するため，3.4 節で説明する逆翻訳モデルのうち Transformer および CNN から生成した 900 万文の擬似データを組み合わせた 1,800 万文の擬似データで事前学習された GEC モデルを用意した．さらに，異なる逆翻訳モデルを使用した場合と比較するため，シードのみが異なる単一の逆翻訳モデルから擬似データを生成し，同様に組み合わせた擬似データで事前学習された GEC モデルを用意した．ここで，全ての逆翻訳モデルに与えた文は同じである．そのため，擬似データを組み合わせた際，擬似誤り文側の種類数は増えているが，訂正文側の種類数は増えていない．

1） 2020 年 7 月 6 日 のダンプデータを用いた．
2） 逆翻訳モデルのシードの違いによる影響を軽減するため，各シードの GEC モデルごとに対応するシードで訓練した逆翻訳モデルを用意した．そして，その対応する逆翻訳モデルから生成した擬似データを GEC モデルの事前学習に使用した．

表 1: それぞれの GEC モデルの訂正性能（シングルモデル/アンサンブルモデル）．上段は逆翻訳モデルごとの性能を表し，下段は擬似データを組み合わせて使用した場合の性能を表す．

	CoNLL-2014			JFLEG	BEA-test		
逆翻訳モデル	Prec.	Rec.	$F_{0.5}$	GLEU	Prec.	Rec.	$F_{0.5}$
なし（ベースライン）	58.5/65.8	31.3/31.5	49.8/54.0	53.0/53.7	52.6/61.4	42.8/42.8	50.2/56.5
Transformer	**65.0**/68.6	**37.6/37.7**	**56.7/59.0**	57.7/58.3	61.1/66.5	49.8/50.7	58.4/62.6
CNN	64.0/68.1	37.4/37.4	56.0/58.5	**57.8/58.4**	**61.9/67.5**	**50.7/51.0**	**59.3/63.4**
LSTM	64.7/**68.8**	36.2/36.4	55.9/58.4	57.0/57.4	61.3/67.1	49.5/49.9	58.5/62.8
Transformer & CNN	65.2/**69.1**	**38.7/39.1**	**57.3/59.9**	**57.9**/58.5	**63.1/67.6**	51.1/51.1	**60.2/63.5**
Transformer & Transformer	65.5/68.3	37.9/38.0	57.2/58.9	57.5/58.0	63.0/67.0	51.0/50.7	**60.2**/63.0
CNN & CNN	**65.6/69.1**	38.2/38.7	**57.3**/59.8	**57.9/58.6**	61.9/67.1	**51.4/51.6**	59.5/63.3

3.4 逆翻訳モデル

GEC において逆翻訳を使用している研究が用いているモデルを参考にして，Transformer，CNN，および LSTM を選択した．また，全てのモデルの実装は fairseq にあるものを使用した．さらに，多様な誤りを生成するために，デコード時に Xie ら [9] によって提案されたノイズ付きビームサーチを使用した．本手法では，ビームサーチ時に，毎ステップごとの各仮説のスコアに $r\beta_{\text{random}}$ をノイズとして加える．ここで，r は区間 $[0, 1]$ の一様分布からランダムに選択される値であり，β_{random} はノイズの大きさを調節するためのハイパーパラメータである．本実験においては，Transformer では $\beta_{\text{random}} = 8$，CNN では $\beta_{\text{random}} = 10$，LSTM では $\beta_{\text{random}} = 12$ とした[3)]．

4 実験結果

4.1 全体の訂正性能

逆翻訳モデルごとの訂正性能　表 1 の上段に，逆翻訳モデルごとの訂正性能を示す．表 1 より，評価データによって，優れている逆翻訳モデルが異なることが分かる．例えば，CoNLL-2014 では Transformer を使用した場合に最も性能が高く，JFLEG および BEA-test では CNN を使用した場合に最も性能が高いことが分かる．また，BEA-test では，Transformer よりも LSTM を使用した場合の方が，$F_{0.5}$ 値が高いことが分かる．したがって，GEC モデルとして高い性能を持つ Transformer [12] が，必ずしも逆翻訳モデルとしても優れているとは限らないことが示唆される．

擬似データを組み合わせた場合の訂正性能　表 1 の下段に，擬似データを組み合わせた場合の訂正性能を示す．表 1 より，異なる逆翻訳モデルから生成した擬似データを組み合わせて使用した場合，擬似データを組み合わせない場合よりも一貫して性能が向上していることが分かる．一方で，シードのみが異なる単一の逆翻訳モデルから生成した擬似データを組み合わせた場合，一部の項目において組み合わせない場合よりも性能が低下していることが分かる．例えば，Transformer では，アンサンブルモデルにおける CoNLL-2014 上の $F_{0.5}$ 値で，組み合わせない場合のスコアは 59.0 であるが，組み合わせた場合のスコアは 58.9 である．また，CNN の場合でも同様に，アンサンブルモデルにおける BEA-test 上の $F_{0.5}$ 値で，組み合わせない場合のスコアは 63.4 であるが，組み合わせた場合のスコアは 63.3 である．したがって，異なる逆翻訳モデルから生成した擬似データを組み合わせて使用する方が，シードのみが異なる単一の逆翻訳モデルを使用するよりも，より頑健な GEC モデルが構築されると考えられる．

4.2 誤りタイプ別の訂正性能

逆翻訳モデルごとの訂正性能　表 2 の左側に，シングルモデルにおける BEA-test 上での誤りタイプ別の $F_{0.5}$ 値を示す．表 2 より，Transformer では PRON（代名詞）の誤りの性能が高いことが分かる．また，CNN では PREP（前置詞），VERB:TENSE（時制）および VERB:SVA（主語と動詞の一致）の誤り，LSTM では VERB（動詞）の誤りの性能が高いことが分かる．したがって，逆翻訳モデルごとに誤りタイプ別の訂正傾向が異なると考えられる．

また，Transformer を逆翻訳モデルに使用した場合，PUNCT（句読点）の誤りの性能がベースラインよりも低下している．さらに，CNN と LSTM の場合でも，他の誤りタイプと比較して，PUNCT の誤りはベースラインからの性能の向上幅が小さいこと

3)　事前実験において，BEA-valid 上で最大の $F_{0.5}$ 値となった時の値を使用した．

表 2: シングルモデルにおける BEA-test 上での誤りタイプ別の $F_{0.5}$ 値．頻度が 100 以上である誤りタイプを抜き出した．また，全ての誤りタイプの頻度の合計は 4,882 である．なお，誤りタイプの詳細は文献 [37] を参照せよ．

			逆翻訳モデル					
誤りタイプ	頻度	ベースライン	Transformer	CNN	LSTM	Transformer & CNN	Transformer & Transformer	CNN & CNN
OTHER	697	22.2±1.77	**31.8±0.71**	31.7±0.77	30.6±0.16	**34.2±1.03**	31.8±1.01	31.6±0.74
PUNCT	613	65.6±2.02	64.6±0.42	**67.8±0.83**	67.3±1.83	65.9±1.51	66.0±0.73	**67.8±0.93**
DET	607	53.8±0.71	64.8±1.62	65.0±0.41	**65.2±0.83**	64.8±0.64	**66.7±1.15**	64.7±0.75
PREP	417	48.2±0.55	58.1±0.76	**59.3±0.54**	55.2±1.74	**61.1±0.43**	60.3±0.76	60.3±1.06
ORTH	381	72.7±2.47	77.2±0.50	**78.7±1.50**	78.0±1.95	**79.2±1.25**	78.4±1.28	78.8±0.74
SPELL	315	58.3±3.49	71.0±1.71	71.1±1.45	**71.6±0.50**	**73.3±1.03**	72.5±0.40	71.1±0.49
NOUN:NUM	263	57.8±2.23	**64.4±1.09**	63.7±0.90	63.9±1.35	66.2±0.43	**66.3±0.61**	64.6±1.41
VERB:TENSE	256	43.9±2.35	52.1±1.58	**54.6±0.94**	52.6±0.50	53.7±1.71	54.6±0.64	**54.8±1.27**
VERB:FORM	213	62.0±2.26	66.7±2.63	**67.1±0.46**	66.0±1.60	66.3±0.34	**66.9±1.54**	66.6±1.01
VERB	196	32.5±3.41	36.0±1.18	36.3±0.91	**39.7±3.05**	**42.7±3.83**	39.0±0.76	38.2±0.98
VERB:SVA	157	66.1±1.38	73.7±3.00	**75.6±0.86**	73.8±2.51	75.1±1.04	**76.3±1.20**	74.3±0.44
MORPH	155	54.0±2.03	61.9±1.97	**63.8±1.23**	**63.8±0.53**	64.5±0.62	**66.3±1.26**	63.8±2.84
PRON	139	43.8±2.00	**53.0±2.79**	51.8±0.14	49.6±1.93	**53.3±1.10**	52.7±2.75	**53.3±0.46**
NOUN	129	19.7±2.04	**31.4±0.62**	30.2±2.39	30.5±2.17	**35.9±2.90**	34.5±1.48	32.8±2.80

が分かる．したがって，逆翻訳による擬似データを使用した場合，PUNCT の誤りは性能を改善させることが難しい誤りタイプであると考えられる．

擬似データを組み合わせた場合の訂正性能　表 2 の右側に，擬似データを組み合わせた場合のシングルモデルにおける BEA-test 上での誤りタイプ別の $F_{0.5}$ 値を示す．表 2 より，PUNCT，VERB:TENSE および VERB:FORM（不定詞・動名詞・分詞の用法）を除く誤りタイプにおいて，異なる逆翻訳モデルから生成した擬似データを組み合わせた場合の方が，シードのみが異なる単一の逆翻訳モデルを使用した場合の少なくとも一方よりも性能が高いことが分かる．そのため，異なる逆翻訳モデルから生成した擬似データを使用した場合，シードのみが異なる単一の逆翻訳モデルを使用した場合に対して，性能が向上するあるいは補間する性能を持つと考えられる．

また，OTHER（その他）の誤りについて，シードのみが異なる単一の逆翻訳モデルから生成した擬似データを組み合わせた場合，組み合わせない場合と比較して，性能が向上していないことが分かる．一方で，異なる逆翻訳モデルから生成した擬似データを組み合わせた場合は，組み合わせない場合よりも性能が向上している．したがって，異なる逆翻訳モデルを使用することにより，より多様な誤りタイプを訂正することが可能になると考えられる．

シードの違いによる影響　ここでは，逆翻訳モデルのシードの違いによる影響について検討する．表 2 より，異なる逆翻訳モデルから生成した擬似データを組み合わせた場合よりも，シードのみが異なる単一の逆翻訳モデルを使用した場合の方が，高い性能になる誤りタイプがあることが分かる．我々は，この理由の一つとして，逆翻訳モデルのシードのみを変えた場合でもある程度性能にばらつきがあるためであると考える．例えば，逆翻訳モデルに Transformer を使用した場合，DET（限定詞）の誤りの標準偏差は 1.62 と比較的高い．またこの時，Transformer & CNN よりも，Transformer & Transformer の方が高い性能になっている．このように，ある程度性能にばらつきがある誤りタイプでは，シードのみが異なる単一の逆翻訳モデルを使用した場合でも，異なる逆翻訳モデルを使用した場合と比べて，性能が向上する可能性があると考えられる．

5　おわりに

本研究では，逆翻訳モデルごとの GEC モデルの訂正傾向を調査した．その結果，逆翻訳モデルごとに誤りタイプ別の訂正傾向が異なることが分かった．また，異なる逆翻訳モデルから生成した擬似データを組み合わせた場合，シードのみが異なる単一の逆翻訳モデルから生成した擬似データを組み合わせた場合に対して，性能が向上するあるいは補間する性能を持つことを確認した．

謝辞

Lang-8 のデータを使用したことについて，株式会社 Lang-8 の喜洋洋氏に感謝申し上げます．

参考文献

[1] Zheng Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In *NAACL*, pp. 380–386, 2016.

[2] Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. A Nested Attention Neural Hybrid Model for Grammatical Error Correction. In *ACL*, pp. 753–762, 2017.

[3] Shamil Chollampatt and Hwee Tou Ng. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. In *AAAI*, pp. 5755–5762, 2018.

[4] 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 *NAACL*, pp. 156–165, 2019.

[5] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. In *ICML*, pp. 1243–1252, 2017.

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

[7] Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. Learning to combine Grammatical Error Corrections. In *BEA*, pp. 139–148, 2019.

[8] Philipp Koehn and Rebecca Knowles. Six Challenges for Neural Machine Translation. In *WNMT*, pp. 28–39, 2017.

[9] Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction. In *NAACL*, pp. 619–628, 2018.

[10] Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. Corpora Generation for Grammatical Error Correction. In *NAACL*, pp. 3291–3301, 2019.

[11] Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data. In *BEA*, pp. 252–263, 2019.

[12] Shun Kiyono, Jun Suzuki, Tomoya Mizumoto, and Kentaro Inui. Massive Exploration of Pseudo Data for Grammatical Error Correction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, pp. 2134–2145, 2020.

[13] 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 *ACL*, pp. 4248–4254, 2020.

[14] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving Neural Machine Translation Models with Monolingual Data. In *ACL*, pp. 86–96, 2016.

[15] Tao Ge, Furu Wei, and Ming Zhou. Fluency Boost Learning and Inference for Neural Grammatical Error Correction. In *ACL*, pp. 1055–1065, 2018.

[16] Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. Improving Grammatical Error Correction with Data Augmentation by Editing Latent Representation. In *COLING*, pp. 2202–2212, 2020.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, pp. 5998–6008, 2017.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[19] Phu Mon Htut and Joel Tetreault. The Unbearable Weight of Generating Artificial Errors for Grammatical Error Correction. In *BEA*, pp. 478–483, 2019.

[20] Yikang Shen, Zhouhan Lin, Chin wei Huang, and Aaron Courville. Neural Language Modeling by Jointly Learning Syntax and Lexicon. In *ICLR*, 2018.

[21] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. In *ICLR*, 2019.

[22] Max White and Alla Rozovskaya. A Comparative Study of Synthetic Data Generation Methods for Grammatical Error Correction. In *BEA*, pp. 198–208, 2020.

[23] 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 *BEA*, pp. 213–227, 2019.

[24] Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. Improving Grammatical Error Correction with Machine Translation Pairs. In *EMNLP Findings*, pp. 318–328, 2020.

[25] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 Shared Task on Grammatical Error Correction. In *BEA*, pp. 52–75, 2019.

[26] Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A New Dataset and Method for Automatically Grading ESOL Texts. In *ACL*, pp. 180–189, 2011.

[27] 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 *IJCNLP*, pp. 147–155, 2011.

[28] Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. Tense and Aspect Error Correction for ESL Learners Using Global Context. In *ACL*, pp. 198–202, 2012.

[29] Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *BEA*, pp. 22–31, 2013.

[30] Sylviane Granger. The computerized learner corpus: a versatile new source of data for SLA research. In *Learner English on Computer*, pp. 3–18. 1998.

[31] Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. Developing an Automated Writing Placement System for ESL Learners. *Applied Measurement in Education*, Vol. 31, No. 3, pp. 251–267, 2018.

[32] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *ACL*, pp. 1715–1725, 2016.

[33] 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 *CoNLL*, pp. 1–14, 2014.

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

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

[36] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground Truth for Grammatical Error Correction Metrics. In *ACL-IJCNLP*, pp. 588–593, 2015.

[37] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *ACL*, pp. 793–805, 2017.

[38] Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments. In *COLING*, pp. 825–835, 2016.

[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 *NAACL*, pp. 48–53, 2019.

[40] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[41] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. In *ICML*, pp. 4596–4604, 2018.