教師付き注意機構を用いた文法誤り訂正

石井 宏道　　田村 晃裕　　二宮 崇
愛媛大学 大学院理工学研究科 電子情報工学専攻
{ishii@ai., tamura@, ninomiya@}cs.ehime-u.ac.jp

1 はじめに

文法的な誤りを含む文（誤り文）を正しい文（訂正文）に自動的に訂正する文法誤り訂正技術は，外国語の学習支援などに役立つことから，盛んに研究されている．これまで様々な手法が提案されているが，近年では，ニューラルネットワークを用いた手法が最高精度を達成し，主流となっている．ニューラルネットワークによる文法誤り訂正モデルの中で特に性能の良いモデルが Transformer[10] に基づくモデルである．Transformer に基づくモデルの特徴の一つが，同一文内（誤り文内あるいは訂正文内）の単語間の関連を捉える自己注意機構と，訂正文の単語を生成する際に誤り文のどの単語に着目するか（訂正文の単語と誤り文の単語の関連）を捉えるエンコーダ・デコーダ注意機構を有していることである．通常，これらの注意機構で捉える単語間の関連は自動的に学習される．

Chollampatt and Ng[2] は，CNN に基づく文法誤り訂正モデルと LSTM に基づく文法誤り訂正モデルで自動的に捉えられたエンコーダ・デコーダ注意を比較・分析し，訂正文の各単語の注意は，誤り文の中の対応する単語にむけるよりも，誤り文の中で対応する単語の周辺単語にむけた方が文法誤り訂正に貢献すると考察している．

本稿は，Transformer に基づく文法誤り訂正モデルの一つである Copy-Augmented Transformer[12] において，複数ヘッドのエンコーダ･デコーダ注意機構の一つのヘッドに対して，訂正文の各単語の注意を誤り文の中で対応する単語の周辺単語にむけさせる制約を与えて学習する手法を提案する．提案手法により，誤りを周辺単語に着目して訂正するモデルが学習できると期待される．CoNLL-2014 テストセット [8] を用いた文法誤り訂正の評価実験により，訂正元の単語の周辺単語に注意をむけさせる制約を Copy-Augmented Transformer に導入することで，$F_{0.5}$ が 0.76 ポイント向上することを確認できた．

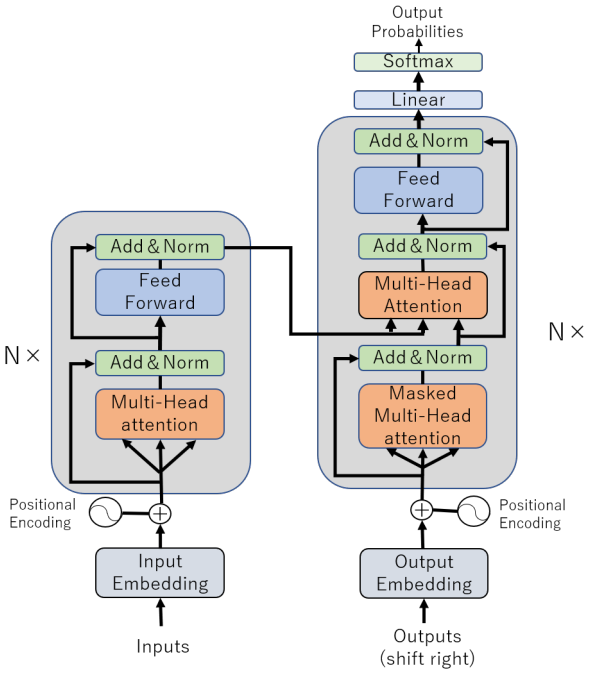

図 1: Transformer の概要図

2 Copy-Augumented Transformer

本節では，提案モデルのベースとなる Copy-Augumented Transformer[12] を説明する．このモデルは Transformer モデルに Copy Mechanism と事前学習を導入することで高い誤り訂正性能を実現している．以降では，2.1 節で Transformer について説明し，2.2 節で Copy Mechanism，2.3 節で事前学習の方法を説明する．

2.1 Transformer

Transformer は，入力系列 $X = (x_1, x_2, \cdots, x_n)$ を中間表現 $Z = (z_1, z_2, \cdots, z_n)$ に変換するエンコーダと中間表現 Z から出力系列 $Y = (y_1, x_2, \cdots, y_m)$ を生

成するデコーダを組み合わせたエンコーダ・デコーダモデルである．Transformer の概要を図 1 に示す．

エンコーダとデコーダは，それぞれ N 個スタックされたエンコーダ層とデコーダ層で構成される．エンコーダ層は，複数ヘッドの自己注意機構と単語位置毎の全結合層の二つのサブ層で構成される．デコーダ層は，複数ヘッドの自己注意機構と単語位置毎の全結合層，複数ヘッドのエンコーダ・デコーダ注意機構の三つのサブ層によって構成される．各サブ層の間は，残差接続と層正規化が行われる．複数ヘッド自己注意機構と複数ヘッドエンコーダ･デコーダ注意機構の各ヘッドでは，式 (1) の計算を行う，スケール化内積注意機構が用いられる．

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

ここで，Q,K,V は，それぞれ query，key，value を表し，エンコーダもしくはデコーダの内部状態である．また，d_k は内部状態の次元のサイズを表す．スケール化内積注意機構では，Q と K の内積を計算することで要素間の関連度を算出する．そして，算出した値に softmax 関数を適用して V と掛け合わせることで，要素間の関連の強さを重みとする荷重和表現を求める．自己注意機構では，Q,K,V として同一の入力源（エンコーダ内の内部状態またはデコーダ内の内部状態）を用いることで同一文内の単語間の関連の強さを計算できる．エンコーダ・デコーダ注意機構では，Q としてデコーダの内部状態，K と V としてエンコーダの最終出力を用いることで，入力文の各単語と出力単語との関連の強さを計算できる．その後，各ヘッドで得られた表現を結合し，単語埋め込み次元に線形変換する．

$$MultiHead(Q,K,V) = Concat(h_1,...,h_h)W_O \quad (2)$$

$$h_i = Attention(Q_i,K_i,V_i) \quad (3)$$

ここで，W_O は重み行列である．

N 層目のデコーダ層の出力が算出されたら，その出力を語彙数次元に線形変換し，softmax 関数を適用することで出力単語に対する確率分布を算出する．そして，算出した確率分布に基づいて出力文を生成する．

モデル学習時に使用する損失関数は下記の式 (4) の通りである．

$$-\sum_{i=1}^{D} \log P(Y^i \mid X^i;\theta) \quad (4)$$

ここで，D は誤り訂正モデルの学習データ数であり，i 番目の学習データが (X^i,Y^i) である．

2.2 Copying Mechanism

Copying Mechanism[9] は，デコーダで文を生成する際に，単語を生成するか入力文の単語をコピーするかを逐次的に決める機構である．具体的には，t 番目の出力単語を以下の式 (5) の確率にしたがって決定する．

$$\begin{aligned} P(y_t|X) &= (1-\alpha_t^{copy}) \times P^{gen}(y_t|X) \\ &\quad +\alpha_t^{copy} \times P^{copy}(y_t|X) \end{aligned} \quad (5)$$

P^{gen} は Transformer のデコーダで算出された出力単語に対する確率であり，P^{copy} は入力文から単語をコピーする確率である．また，α_t^{copy} は単語を生成するかコピーするかを調整する重みである．P^{copy} 及び α_t^{copy} は，以下ように，エンコーダ・デコーダ注意に基づき算出する．

$$q_t = h_t^{trg}W_Q^T,\ K = H^{src}W_K^T,\ V = H^{src}W_V^T \quad (6)$$

$$A_t = q_t^T K \quad (7)$$

$$P^{copy}(y_t|X) = softmax(A_t) \quad (8)$$

$$\alpha_t^{copy} = sigmoid(W^T \sum(A_t^T \cdot V)) \quad (9)$$

ここで，h_t^{trg} はタイムステップ t のデコーダの内部状態，H^{src} はエンコーダの内部状態系列 $(h_1^{src},...,h_n^{src})$，$W_Q$，$W_K$，$W_V$，$W$ は，それぞれ重み行列である．

2.3 事前学習

学習データ量を補うためにノイズ除去自己符号化器による事前学習を行う．具体的には，One Billion Word Benchmark データ [1] の文に対して，10%の確率で文中の単語を削除したり，単語を挿入したり，文中の単語と辞書の単語と入れ替えたり，単語の位置関係を入れ替えたりすることで疑似誤り文を作成する．そして，元の文と疑似誤り文の対から copy-augmented Transformer のパラメーターを事前学習する．

3 提案手法

本節では，Transformer の複数ヘッドエンコーダ・デコーダ注意機構の一つのヘッドに対して，訂正文の各単語の注意を，誤り文の中で対応する単語の周辺単語にむけさせる制約を与えて学習する手法を提案する．図 2 にエンコーダ・デコーダ注意に対する制約のため

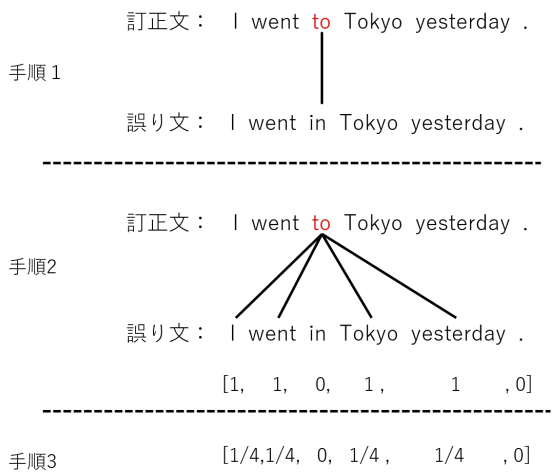

図 2: エンコーダ・デコーダ注意に対する制約の教師データ作成例

の教師データの作成例を示す．提案手法では，まず，誤り文と訂正文の単語の対応関係をアライメントツールを用いて解析する（手順 1）．その後，手順 1 によって得られた単語アライメント結果から訂正文の各単語に対して，対応する誤り文の単語の前後 2 単語を特定する（手順 2）．そして，訂正文の単語毎に，手順 2 で特定された単語に対しては「1/特定された単語数」の確率を与え，誤り文中のその他の単語に対する確率は 0 とする（手順 3）．例えば，図 2 の訂正文中の単語「to」に対しては，誤り文の中で対応する単語は「in」の 1 単語のみであり，その前後 2 単語は「I」,「went」,「Tokyo」,「yesterday」の 4 単語である．したがって，それら 4 単語に対しては 0.25（=1/4）の確率を与え，誤り文中のその他の単語（「in」と「.」）に対する確率は 0 とする．

このように作成した訂正文の各単語に対する確率分布を複数ヘッドエンコーダ・デコーダ注意機構の一つのヘッドの教師データとして使い，誤り訂正モデルを学習する．具体的には，誤り訂正モデルを学習する際，式 (4) の損失関数の代りに，エンコーダ・デコーダ注意機構に対する誤差を加えた，下記式 (10) の損失関数を用いる．

$$-\sum_{i=1}^{D} \log P(Y^i \mid X^i; \theta) + \lambda \times \Delta(\alpha^i, \hat{\alpha^i}; \theta) \qquad (10)$$

ここで，λ はハイパーパラーメータであり，Δ はエンコーダ・デコーダ注意機構で捉えた単語の対応関係 α^i と教師データの単語間対応関係 $\hat{\alpha^i}$ との誤差であり，下記の式 (11) で求められる．

$$\Delta(\alpha^i, \hat{\alpha^i}; \theta) = -\sum_{m}\sum_{n} \hat{\alpha^i}_{m,n} \times \log \alpha(\theta)^i_{m,n} \qquad (11)$$

ここで，$\hat{\alpha^i}_{m,n}$ と $\alpha(\theta)^i_{m,n}$ は単語 m と単語 n の関連を表す確率値であり，具体的には，$\hat{\alpha^i}_{m,n}$ はエンコーダ・デコーダ注意に対する制約の教師データ作成手順 3 で求めた確率，$\alpha(\theta)^i_{m,n}$ はエンコーダ・デコーダ注意機構で算出する式 (1) の荷重和表現における重みに相当する．

4 実験

4.1 実験データ

学習データは，NUS Corpus of Learner English (NUCLE)[4]，Lang-8 learner Corpus[7]，FCE[11] を合わせた約 120 万文対を使用した．開発データは，CoNLL-2013 テストデータ [4] を使用し，評価データとして CoNLL-2014 テストセットを使用した．それぞれのデータに対する前処理は，Zhao ら [12] に従った．

4.2 実験設定

本実験では，Copy-Augmented Transformer[12] をベースラインとして提案手法と性能比較する．ベースラインのモデルは Zhao らによる実装[1]を用いた．提案手法はベースラインに対して 3 節で説明したエンコーダ・デコーダ注意に対する制約を与えて学習したモデルである．ベースラインモデルと提案モデル共に，エンコーダ層とデコーダ層はそれぞれ 6 層スタックし，ヘッド数は 8，内部状態の次元数は $d_{model} = 512$，$d_{ff} = 4096$ とした．また，最適化は Nesterovs Accelerated Gradient を使用し，学習率は 0.02，重み減衰は 0.5 とした．edit-weighted MLE を用い $\Lambda = 1.2$ にした．デコードの際は，窓幅 12 のビーム探索を用いた．

提案手法のハイパーパラメータ λ は 0.05 とし，制約は 5 層目のエンコーダ・デコーダ注意機構に導入した．また，制約を与える際に使う教師データ作成のためのアライメントツールは GIZA++を用いた．

誤り訂正性能は，MaxMatch (M^2) scorer [3] で計算された $F_{0.5}$ 値により評価した．

[1]https://github.com/zhawe01/fairseq-gec

表 1: 実験結果

手法	Precision(%)	Recall(%)	$F_{0.5}$(%)
ベースラインモデル	68.30	37.93	58.87
提案手法	69.10	38.02	**59.60**
Chollampatt and Ng (2018)[2]	60.9	23.7	46.4
Junczys-Downmunt el at (2018)[6]	-	-	53.0
Grundkiewicz and Junczys-Dowmunt (2018)[5]	66.8	34.5	56.3
Lichtarge et al. (2019) [6]	65.5	37.1	56.8
Zhao et al. (2019) (ベースラインモデルの論文値)[12]	68.97	36.98	58.80
Zhao et al. (2019) (ベースラインモデル + マルチタスク学習の論文値)[12]	67.74	40.62	59.76

4.3 実験結果

実験結果を表 1 に示す．表は，CoNLL-2014 テストセットに対するベースラインモデル，提案手法およびアンサンブルを用いない既存手法の精度を表す．表 1 より，提案手法の性能がベースラインの性能よりも $F_{0.5}$ で 0.73 ポイント上回った．この結果より，訂正文の各単語の注意を誤り文の中で対応する単語の周辺単語にむけさせる制約を与えてエンコーダ・デコーダ注意機構を学習することで誤り訂正性能を改善できることが実験的に確認できた．また，表は比較のために，ensemble を用いない既存手法の精度も示している．本研究のベースラインモデルおよび提案手法はマルチタスク学習を行っていないため，Zhao らのマルチタスク学習を用いた手法 [12] に比べて精度が劣っている．

5 おわりに

本稿では，訂正文中の各単語の注意を誤り文の中で対応する単語の周辺にむけるように制約を与えて，エンコーダ・デコーダ注意機構を学習する手法を提案した．実験より，Copy-Augmented Transformer の複数エンコーダ・デコーダ注意機構の一つのヘッドに対して提案の制約を与えて学習することにより，$F_{0.5}$ 値が 0.76 ポイント向上することを確認した．今後は，他のデータセットや他の誤り訂正モデルに対しても提案手法の有効性を確認したい．

6 謝辞

本研究成果は，国立研究開発法人情報通信研究機構の委託研究により得られたものである．ここに謝意を表する．

参考文献

[1] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*, Vol. abs/1312.3005, , 2013.

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

[3] Daniel Dahlmeier and Hwee Tou Ng. Better evaluation for grammatical error correction. In *Proc. of NAACL-HLT*, pp. 568–572, 2012.

[4] 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*, pp. 22–31, 2013.

[5] Roman Grundkiewicz and Marcin Junczys-Dowmunt. Near human-level performance in grammatical error correction with hybrid machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 284–290, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[6] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 595–606, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[7] 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.

[8] 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*, pp. 1–14, 2014.

[9] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL*, pp. 1073–1083, 2017.

[10] 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in NIPS*, pp. 5998–6008. 2017.

[11] Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading ESOL texts.

[12] 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. *CoRR*, Vol. abs/1903.00138, , 2019.