パイプライン処理によるニューラル英語文法誤り検出と訂正

金子 正弘　小町 守

首都大学東京

kaneko-masahiro@ed.tmu.ac.jp, komachi@tmu.ac.jp

1 はじめに

近年，ニューラルネットワークを用いた文法誤り訂正の研究が盛んである [1]．文法誤り訂正は，入力として文法的に誤っている可能性がある文を受け取り，その入力文に対して文法的に正しくなるように訂正を行う．文法誤り訂正をうまく行うには，原文の誤った箇所が文法的に正しく書き換わっている必要がある．そのため，文法誤り訂正では訂正するべき箇所を特定することも重要になる．

これまでの文法誤り訂正器のうち，分類器を用いた手法 [2] では明示的に検出の情報を訂正の際に考慮しているが，ニューラルネットワークを用いた文法誤り訂正手法では，明示的に誤り検出の情報が使われていないだけでなく議論もされていない．一方で，誤り訂正器は検出器と比較して必ずしも誤り検出がうまく行えているとは言えない [3]．そのため，誤り訂正の適合率を上げるためには，補助情報として誤り検出の情報を使うことが有効であると考えられる．

そこで，本研究では入力文のどの箇所が誤っているかをパイプライン処理によってニューラル文法誤り訂正器に与えることで入力文の誤り箇所特定性能の向上を図る．具体的には，誤り訂正器とは独立した検出器の検出結果を素性として入力文の単語と同時に訂正器のエンコーダに入力するモデルを提案する．このモデルは入力文の誤り検出の情報をシンプルに考慮することが可能である．

英語文法誤り訂正の実験では，誤り検出の情報を素性として用いることが文法誤り訂正の精度向上につながることがわかった．そして，検出結果を素性として学習することで，誤り訂正器の検出精度が誤り検出器以上に向上することを示した．

2 パイプライン処理による文法誤り検出と訂正

本研究では，文法誤り訂正器として Luong らのモデル [4] を使う．誤り訂正器のエンコーダに素性として誤り検出結果を入力し，単語単位のアテンションを用いて誤り訂正を行う．誤り検出結果をエンコーダに加えることで，誤り訂正の適合率の向上が期待できる．

誤り検出　文法誤り検出タスクにおいて高い精度を出している Bidirectional LSTM (Bi-LSTM) を誤り検出器として用いる [3]．入力文 $S = (w_1, w_2, \cdots, w_N)$ の各単語 w_i は単語ベクトル $e_i \in \mathbb{R}^{d_e \times 1}$ に変換される．N は文長であり，d_e は単語ベクトルの次元である．単語ベクトルから順方向の隠れ層 $\overrightarrow{h_i} \in \mathbb{R}^{d_h \times 1}$ と逆方向の隠れ層 $\overleftarrow{h_i} \in \mathbb{R}^{d_h \times 1}$ を作成する．d_h は隠れ層の次元とする．$\overrightarrow{h_i}$ と $\overleftarrow{h_i}$ を連結することで最終的な隠れ層 $h_i^{(lstm)} \in \mathbb{R}^{2d_h \times 1}$ を獲得する．隠れ層 $h_i^{(lstm)}$ を以下のように線形変換し，ソフトマックス関数を用いて正誤タグの確率分布 $p_i \in \mathbb{R}^{t \times 1}$ を獲得する．t はタグのサイズであり 2 である．

$$p_i = \mathrm{softmax}(W_h h_i^{(lstm)} + b_h) \tag{1}$$

$W_h \in \mathbb{R}^{v \times d_h}$ は重み行列であり，$b_h \in \mathbb{R}^{v \times 1}$ はバイアスである．v は語彙サイズの次元数である．

誤り訂正　学習済みの文法誤り検出器を使い入力文 S の各単語 w_i に対して誤り検出を行う．さらに，入力文 S の各単語 w_i から単語ベクトル $v_i \in \mathbb{R}^{d_v \times 1}$ を作成する．そして，単語 w_i に対する誤り検出の出力 p_i を単語ベクトル e_i と連結し $x_i = [e_i; p_i] \in \mathbb{R}^{(d_e+t) \times 1}$ として誤り訂正器に入力する．これにより，文法誤り訂正器は誤り検出の情報を考慮して学習することが可能となる．v_i からエンコーダの隠れ層 $h_i^{(enc)} \in \mathbb{R}^{d_h \times 1}$ を Bi-LSTM の隠れ層と同じように計算する．

デコーダの初期状態 $h_0^{(dec)} \in \mathbb{R}^{d_h \times 1}$ はエンコーダから算出された最終 LSTM ユニット $h_N^{(enc)}$ とそのメモリセルで初期化される．j 番目の単語を出力するためにエンコーダの隠れ層の重み付き和 a_j を計算する．

$$a_j = \sum_{i=1}^{N} \alpha_j(i) h_i^{(enc)} \quad (2)$$

$$\alpha_j(i) = \frac{\exp(h_i^{(enc)} W_a h_j^{(dec)})}{\sum_{k \neq j}^{N} \exp(h_k^{(enc)} W_a h_j^{(dec)})} \quad (3)$$

$$\tilde{h}_j^{(dec)} = \tanh(W_c[h_j^{(dec)}; a_j] + b_c) \quad (4)$$

$\alpha_j(i)$ は j 番目の出力をする際の，エンコーダの i 番目の隠れ層の寄与率を示している．$W_a \in \mathbb{R}^{(2d_h+dv)\times v}$ と $W_c \in \mathbb{R}^{d_h \times (v+d_h)}$ は重み行列であり，$b_c \in \mathbb{R}^{d_h \times 1}$ はバイアスである．デコーダの最終的な隠れ層 $\tilde{h}_j^{(dec)} \in \mathbb{R}^{d_h \times 1}$ を用いることで訂正を行う．

$$p(y_j \mid y_{<j}, S) = \mathrm{softmax}(W_w \tilde{h}_j^{(dec)} + b_w) \quad (5)$$

$W_w \in \mathbb{R}^{d_w \times d_h}$ は重み行列であり，$b_w \in \mathbb{R}^{d_w \times 1}$ はバイアスである．d_w は語彙サイズである．

3 英文法誤り訂正実験

3.1 データ

学習データとして NUCLE，FCE-public，Lang-8 と AWES を用いた．そして，開発データとして CoNLL-13，評価データとして CoNLL-14 を用いた．

***FCE-public* データセット.** FCE-public データセットは文法誤り訂正における最も有名な英語学習者コーパスの 1 つで，文法誤りの種類に基づいてタグ付けがされている．

***NUCLE* と *CoNLL*.** さらに，CoNLL-13 [5]，CoNLL-14 [6] の共通タスクのデータと NUS Corpus of Learner English (NUCLE) [5] を用いる．含まれている文法誤りは，英語を母語とするプロの英語教師によって訂正とアノテーションがされている．

***Lang-8* コーパス.** Lang-8 英語学習者コーパスは，英語学習者によって書かれた英文を人手で訂正した 100 万文以上のデータである [7]．誤りの種類はアノテートされていない．

***AWES*.** Automated Evaluation of Scientific Writing 2016 の共通タスクのデータである [8]．他のコーパスと比較して原文と正解文の編集距離が小さいという特徴がある [9] ため，学習データとしては訂正されている文だけを用いた．

前処理としてコメントや英語と記号以外の文字を含む文と 50 単語以上の文を除外した．さらに，訂正前と訂正後で文が分割される文も除外した．そして，検出のための正誤タグは動的計画法を用いて付与した．結果として，NUCLE: 53,342 文，FCE: 26,346 文，AWES: 508,071 文そして Lang-8: 573,314 文からなる合計 1,123,213 文を抽出した．そして，開発セットとして CoNLL-13: 1,381 文，テストセットとして CoNLL-14: 1,312 文を用いた．

3.2 ハイパーパラメータ

誤り検出器の Bi-LSTM の埋め込み層と隠れ層の次元は予備実験の結果両方とも 300 とした．語彙サイズは 40,000 とした．学習には ADAM を使い，学習率は 0.001 とした．重み行列は $[-0.25, 0.25]$ の一様乱数で初期化した．ミニバッチサイズは 128，勾配の大きさは 5 にクリップした．単語ベクトルは Google News で学習された word2vec [1]を使い初期化した．

誤り訂正器の埋め込み層と隠れ層の次元は両方とも 512 とした．語彙サイズは 40,000 とした．学習には SGD を使い，学習率は 1.0 とした．重み行列は $[-0.1, 0.1]$ の一様乱数で初期化した．ミニバッチサイズは 128，勾配の大きさは 1 にクリップした．エンコーダとデコーダの両 LSTM に対して係数 0.2 の dropout を適用した．テスト時には幅 5 のビーム探索を行い，さらに未知語を出力した場合アテンションの寄与率がもっとも高い入力文の単語と置き換えた．

3.3 評価尺度

文法誤り訂正の評価としては，M^2 スコアと GLEU を使う．M^2 スコア [10] は CoNLL-13 と CoNLL-14 の共通タスクで使われた評価尺度であり，原文と出力文を動的計画法で比較し F 値を算出する．一般的に，再現率よりも適合率を重視した $F_{0.5}$ が使われる．GLEU [11] は原文と出力文に存在し正解文に存在しない文字列を減点する手法であり，人手評価と高い相関がある．

3.4 実験結果

表 1 は各モデルの CoNLL-14 に対する誤り訂正結果である．まず，誤り検出結果を素性として用いるアテンションモデルにより誤り訂正結果が改善している

[1] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

表 1: CoNLL-14 データに対する訂正精度

モデル	P	R	$F_{0.5}$	GLEU
アテンションモデル	27.21	**24.25**	25.65	47.88
アテンションモデル＋検出素性	**41.73**	19.06	**26.17**	**58.41**
アテンションモデル＋検出素性（正解）	28.42	26.81	28.08	60.34

表 2: CoNLL-14 データに対する検出精度

モデル	P	R	$F_{0.5}$
Bi-LSTM	**44.05**	8.74	24.37
アテンションモデル	22.77	**26.32**	23.40
アテンションモデル＋検出素性	28.39	18.87	**25.78**

ことを示す．誤り検出結果を用いないアテンションモデル（ベースライン）と比較して誤り検出素性ありのほうが $F_{0.5}$ と GLEU が高いことがわかる．この結果から，誤り検出の情報を誤り訂正器に使うことが精度向上につながることがわかる．

最後のモデルは，誤り検出結果を素性として用いる提案手法に対して正解検出タグを与えた誤り訂正結果であり，提案手法の精度の上界を示している．ベースラインと比較して適合率と再現率の両方が向上している．このことから，誤り検出器の精度を向上させることで誤り訂正器の適合率と再現率の両方の改善が見込めることがわかる．

4 考察

4.1 訂正箇所検出の精度向上

検出器の予測結果を素性として用いることで，誤り訂正器の適合率が上がった．そのため，提案手法の誤り検出精度について調査する．

表 2 は，Bi-LSTM を用いた検出器，誤り検出結果を用いないアテンションモデルと誤り検出の結果を素性として用いるアテンションモデルの検出結果[2]である．ベースライン（検出素性を用いないアテンションモデル）と比較して提案手法（アテンションモデル＋検出素性）の $F_{0.5}$ が向上していることがわかる．これは，適合率の高い検出器の誤り検出の情報を素性として考慮することで，不確実な誤り訂正出力を抑制できたからではないかと考えられる．

4.2 英文法誤り訂正における出力分析

最後に誤り検出の予測結果を素性として用いた誤り訂正器の特徴を分析するために，素性を用いない訂正器と素性を用いた訂正器のそれぞれの誤った出力と両方の訂正器が誤った出力の 3 つを分析する．表 3 は誤り検出結果を用いないアテンションモデルと誤り検出の結果を素性として用いるアテンションモデルの CoNLL-14 データに対する出力例である．1 列目が原文，2 列目は検出器が原文中の対象単語が文法的に誤っていると予測した確率，3 列目が正解文，4 と 5 列目はそれぞれの訂正器の出力である．実際の訂正器には正誤の確率分布のベクトルが素性として入力されている．太字は正しく訂正された単語であり，イタリックは誤って訂正された単語である．

一番上の出力例では，誤り検出の素性を考慮した訂正器がうまく訂正できている．文法的に誤っている単語に対して誤り検出器は，確率は低いが反応することができている．そして，その低い確率をうまく考慮することで訂正器は文法的に誤った単語だけを書き換えて訂正することができている．一方で，誤り検出の予測を情報として使っていないただのアテンションモデルでは適切に誤り検出ができておらず，文法的に正しい単語を誤って訂正し，訂正すべき単語を訂正できていない．この出力例は，誤り検出の結果を確率ではなく，2 値のタグとして訂正器に与えるのではなく確率分布として与えていたためうまく直せた例だと言える．

中央列の出力例では，誤り検出の情報を考慮していない訂正器がうまく訂正している．提案手法については誤り検出はうまく行えているが誤り訂正が適切に行えていない．誤り訂正器は置換の誤りと判断しているが，実際は挿入の誤りである．これは，誤り検出の際に 2 値のタグを用いており挿入の誤りと削除や置換の誤りが区別できないため起こった訂正ミスだと考えられる．

最後の出力例は，どちらの訂正器も誤りを検出することができず訂正することができていない．素性を用いた訂正器は適合率が高いという特徴があり，そのため今回の誤った箇所は検出できず訂正できていない．これは提案手法の特徴であり適切に誤り検出ができている箇所に関しては正解する可能性が高く，誤り検出されていない箇所に関しては正解する可能性が低い．そして，再現率の高いアテンションモデルも誤り検出できていないことから，これは検出することが難しい誤りでもあったと考えられる．

[2]アテンションモデル 2 つの誤り検出結果は，誤り訂正結果を元に動的計画法により算出した．

表 3: CoNLL-14 データに対する出力例

原文	Nothing is absolute right or wrong .
Bi-LSTM の検出結果	0.0 0.0 0.0002 0.0 0.0 0.0 0.0
正解文	Nothing is **absolutely** right or wrong .
アテンションモデル	Nothing *more* is *absolute* or is wrong .
アテンションモデル＋検出素性	Nothing is **absolutely** right or wrong .
原文	Without facebook , we may lost contact .
Bi-LSTM の検出結果	0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0
正解文	Without facebook , we may **have** lost contact .
アテンションモデル	Without facebook , we may **have** lost contact .
アテンションモデル＋検出素性	Without facebook , we may *lose* contact .
原文	With the risk of being genetically disorder , many individuals have done the decision to undergo genetic testing .
Bi-LSTM の検出結果	0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
正解文	With the risk of **having genetic** disorders , many individuals have **made** the decision to undergo genetic testing .
アテンションモデル	With the risk of *being genetically* disorder , many individuals have *done* the decision to undergo genetic testing .
アテンションモデル＋検出素性	With the risk of *being genetically* disorder , many individuals have *done* the decision to undergo genetic testing .

5 先行研究

本研究と同じようにニューラルネットワークを用いた文法誤り訂正の研究としては，文字単位の入出力を可能にしたアテンションモデル [1] などがある．一方で，これの研究は明示的に誤り検出を行っていない．

文法誤り訂正器に検出を組み合わせた研究としては，分類器を用いた研究 [2] や SMT を用いたパイプライン手法 [12] などがある．一方で，これらの研究はニューラルネットワークを用いた文法誤り訂正器ではない．

6 おわりに

本研究では，文法誤り訂正器に誤り検出の予測結果を素性として用いる手法を提案した．CoNLL-14 を用いた文法誤り訂正の実験を行い，提案手法が誤り訂正の適合率を上げることで誤り訂正の精度向上につながることを示した．さらに，訂正器の検出精度を調べたところ誤り検出以上の $F_{0.5}$ スコアを達成した．

今後の取り組みとしては，検出の情報を素性として訂正器に与えるだけではなく，検出器と訂正器を結合し学習させるモデルについての実験や誤りタイプや編集タグのようなより表現力の高いラベルを使った誤り訂正器の研究などが挙げられる．

参考文献

[1] 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*, 2017.

[2] Alla Rozovskaya and Dan Roth. Building a State-of-the-Art Grammatical Error Correction System. In *TACL*, 2014.

[3] Marek Rei and Helen Yannakoudakis. Compositional Sequence Labeling Models for Error Detection in Learner Writing. In *ACL*, 2016.

[4] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*, 2015.

[5] 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*, 2013.

[6] 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 Shared Task*, 2014.

[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 *IJCNLP*, 2011.

[8] Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. A Report on the Automatic Evaluation of Scientific Writing Shared Task. In *BEA Shared Task*, 2016.

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

[10] Daniel Dahlmeier and Hwee Tou Ng. Better Evaluation for Grammatical Error Correction. In *NAACL*, 2012.

[11] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground Truth for Grammatical Error Correction Metrics. In *ACL*, 2015.

[12] Nadi Tomeh, Nizar Habash, Ramy Eskander, and Joseph Le Roux. A Pipeline Approach to Supervised Error Correction for the QALB-2014 Shared Task. In *ANLP*, 2014.