ニューラルネットワークによる英文空所補充問題の一解法

玉城　悠仁† 　新妻　弘崇† 　太田　学†

† 岡山大学大学院自然科学研究科　〒 700–8530 岡山県岡山市北区津島中三丁目 1 番 1 号
E-mail: {tamaki, niitsuma, ohta}@de.cs.okayama-u.ac.jp

あらまし　機械による誤り検出や誤り訂正によって語学の学習者を支援することが期待されている．特に近年はニューラルネットワークを用いた自然言語処理の研究が盛んに行われており，自然言語の誤り検出や誤り訂正にも応用されている．本稿では英文空所補充問題のニューラルネットワークによる解法を提案する．解法では空所の前後それぞれ 10 単語を LSTM 層への入力とし，2 つの LSTM 層の出力をマージするニューラルネットワークのモデルを作成する．また，学習データを text8 コーパス，テストデータを大学入試センター試験の英文空所補充問題として実験を行ったところ，空所に入る語を選択肢の中から予測する際の正答率は 32.2%であり，学習データに含まれる全単語から予測する場合は 1 問も正答することができなかった．

キーワード　ニューラルネットワーク，LSTM，英文空所補充，fastText

1. はじめに

現在，日本人の多くが第二言語として英語を学んでいる．しかし，英語を母語としない日本人にとって正しい英語の使い分けは難しく，日本人の英作文には様々な誤りがしばしば見られる．そういった学習者を支援するために英文の誤り検出や誤り訂正の研究が行われている．

誤り訂正の主要な手法の一つは統計的機械翻訳 (Satatisitic machine translation: SMT) に基づくものである．誤り訂正を，誤りを含む文から，適切な文への翻訳と考え，SMT の手法を誤り訂正に適用している．Junczys-Dowmunt らは文法誤り訂正の代表的なベンチマークデータセットである CoNLL-2014 Shared Task データセット [1] において，SMT を使った手法で高い性能を示した [2]．SMT の分野では近年，部分的な変換ルールは学習せず，文全体の情報を用いて翻訳する sequence-to-sequence 翻訳において，ニューラルネットワーク (Neural network: NN) を用いた手法が注目されている．その最も単純な翻訳モデルとしてリカレントニューラルネットワーク (recurrent neural network: RNN) を用いた encoder-decoder 翻訳モデルがある．Xie ら [3] はこの encoder-decoder 型のモデルによる文字ベースの文法誤り訂正モデルを作成した．

また，小山田ら [4] は Xie らのモデルを改良し，誤り理由を表す隠れ変数を同時に推定し出力することで誤り訂正の根拠も明示するモデルを提案した．Xie らは誤り検出モデルと誤り訂正モデルを別個に学習し，誤り訂正モデルによる修正を実際に適用するかどうかの判断に利用したが，小山田らは誤り理由の種類の中に “誤っていない” というクラスを便宜的に含めることで，誤り検出と誤り訂正を同時に学習できるようにした．さらに，教師ありデータのサンプルサイズが小さい問題に対応するため Nigam ら [5] の提案した半教師あり学習を適用した．

本稿では，NN による誤り検出や誤り訂正の可能性を探るため，英文の空所補充問題への適用を試み，その性能を評価する．具体的には英文の空所の前後の単語列から空所に入る単語を予測する NN モデルを提案する．

本稿は次のような構成となっている. 2. 節で，関連研究について述べ，3. 節では，提案するモデルの構造と処理ついて説明する．4. 節では，評価実験の内容と結果を示すと共に，その結果について考察する．5. 節では，本稿のまとめと今後の課題について述べる．

2. 関連研究

2.1 ニューラルネットワークを用いた英文空所補充

Hermann ら [6] は与えられた文章とその要約である空所つきクエリが与えられたとき，空所に入る語句を文章中の語句から予測するモデルとして，Deep LSTM Reader，Attentive Reader，Impatient Reader の 3 種類のモデルを提案した．

いずれのモデルも文章とクエリのペアからベクトルを生成し，空所にどの単語が入るかの確率を予測するが，そのベクトルの生成方法がモデルによって異なる．Deep LSTM Reader では，文章とクエリを 1 つにつなげたものを多層の LSTM 層からなるモデルに入力してベクトルを生成する．Attentive Reader および Impatient Reader では，文章とクエリをそれぞれ単層の双方向 LSTM に入力して得られた出力をもとに，attention メカニズムを用いて文章ベクトルを生成し，文章ベクトルとクエリベクトルから最終的なペアのベクトルを出力する．文章のベクトルを表現する際，Attentive Reader ではクエリ全体の attention を用いるのに対し，Impatient Reader ではクエリの 1 単語ごとの attention を用いる．

また，Hermann らはベースラインとして文章中で最も多く現れる語句を空所に入る語句として選択する手法や文章とクエリをそれぞれ構文解析して木構造をもとに空所に入る語句を選択する手法などを用いている．

これらのモデルを用いて Cable News Network (注1) と The Daily

(注1)：http://edition.cnn.com/

Mail(注2) の新聞記事を対象に学習，テストを行ったところ，いずれのモデルもベースラインを上回り，The Daily Mail を対象にした実験では Attentive Reader の正答率が 69.0%となった．

2.2 単語の分散表現

NN で自然言語処理を行う際には，文字や文字列をベクトルとしてモデルへ入力する必要がある．ベクトル化の方法には，文字や単語を対応する ID に置き換える方法や，one-hot 表現を用いる方法の他に，分散表現と呼ばれる数値ベクトルで表現する方法がある．この単語を分散表現で表現する方法には，Mikolov らによって提案された word2vec [7, 8] や Bojanowski らによって提案された fastText [9, 10] などがある．word2vec はコーパス中の単語の前後関係を Skip-gram モデルや CBOW モデルと呼ばれる構造の NN で学習し，その NN の中間層の出力を単語のベクトル表現とする．fastText は word2vec を拡張したもので単語の学習の際に単語そのものだけでなく，その単語のスペルの t つづりの subword も利用している．

3. 提 案 手 法

本稿では NN を用いた英文空所補充問題の解法を提案する．例えば {} が空所部分を表すとして，“We have to find {} to the world’ s environmental problems.” などの入力に対して，空所に入る単語（この例では solutions）を予測するモデルを作成する．本モデルは 1 単語を予測するモデルであり，空所に 2 語以上の単語が入る問題については対象としない．空所補充問題では，空所の前後に答えの根拠となる情報が含まれている．そこで，英文の順方向に次の空所に入る語を予測すると同時に，逆方向に前の空所に入る語を予測し，両者の予測結果をマージして出力する単語ベースの双方向モデルを提案する．3.1 節で本モデルの詳細について述べ，3.2 節でモデルの学習時，3.3 節でモデルのテスト時に行う処理の詳細を述べる．

3.1 モデルのネットワーク構造

提案モデルのネットワーク構造を図 1 に示す．図 1 において Input Layer は入力層，Embedding は embedding 層，LSTM は LSTM 層，Merged はマージ層，Fully-connected は全結合層を表す．また，forward は順方向，reverse は逆方向の予測を表す．

本モデルは空所の前後 t 単語を入力として空所に入る単語を予測するモデルである．具体的には空所に入る単語の分散表現を表すベクトルの各成分を予測する回帰モデルである．全結合層における活性化関数は relu，最適化関数は rmsprop，損失関数の損失は平均二乗誤差とした．

3.2 モデルの学習

空所の前後 t 単語を利用して空所に入る単語を予測するため，各学習データは空所分を含めた $2t+1$ 語の単語列である．これを $\boldsymbol{w} = (w_1, \ldots, w_{2t+1})$ とおく．$\boldsymbol{w}$ 前半 t 単語は順方向の予測に，後半 t 単語は逆方向の予測に用い，中央の単語 w_{t+1} は出力の教師ベクトルに用いる．また，モデルへの入力は割り当てた ID に各単語を変換して得られるベクトルとする．ここで学習データに含まれる単語の種類，すなわち語彙数を N とすると，学習データに含まれない単語を追加した $N+1$ 種類の ID がある．すなわち，単語 w_i に対応する ID を id_i とすると $1 \leqq id_i \leqq N+1$ である．ここで，0 は 3.3 節で述べるテストの際に 0 をパディングに用いるため id としては用いない．以上からモデルへ入力のうち順方向の予測に用いる入力を $\boldsymbol{in}_{\mathrm{f}}$，逆方向の予測に用いる入力を $\boldsymbol{in}_{\mathrm{r}}$ とすると，それぞれ以下のように表される．

$$\boldsymbol{in}_{\mathrm{f}} = (id_1, id_2, \ldots, id_t)$$
$$\boldsymbol{in}_{\mathrm{r}} = (id_{2t+1}, id_{2t}, \ldots, id_{t+2})$$

これらの入力の成分である各 ID は，embedding 層を通して，対応する単語を表すベクトルへと変換される．本来, emmbeding 層ではこの変換の重みを学習によって更新するが，本モデルでは embedding 層の重みの初期値として fastText の出力を利用し，本モデルの学習時に重みを更新しないようにする．以上より，fastText によって出力される単語 w_i の分散表現を表すベクトルを $\boldsymbol{v}_i$，順方向および逆方向の embedding 層の出力をそれぞれ $f(\boldsymbol{in}_{\mathrm{f}})$，$f(\boldsymbol{in}_{\mathrm{r}})$ とすると，以下のように表される．

$$f(\boldsymbol{in}_{\mathrm{f}}) = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_t)$$
$$f(\boldsymbol{in}_{\mathrm{r}}) = (\boldsymbol{v}_{2t+1}, \boldsymbol{v}_{2t}, \ldots, \boldsymbol{v}_{t+2})$$

その後の処理は他の NN のモデルと同様に図 1 に示した構造通りに処理する．embedding 層の出力を LSTM 層への入力とし，順方向の LSTM 層の出力と逆方向の LSTM 層の出力をマージする．その後，マージ層の出力を全結合層の入力として，モデルの最終的な出力を行う．ここで，モデルの出力の教師ベクトルは $\boldsymbol{v}_{t+1}$ とし，モデルの出力と教師ベクトルとの損失が小さくなるように LSTM 層と全結合層の重みが学習によって更新される．

3.3 モデルのテスト

各テストデータは “We have to find {} to the world’ s environmental problems.” のような空所を含む単語列である．空所の前 t 単語と後 t 単語がそれぞれ id に変換されモデルへの入力となるが，t 語に満たない場合には入力ベクトルの先頭を 0 でパディ

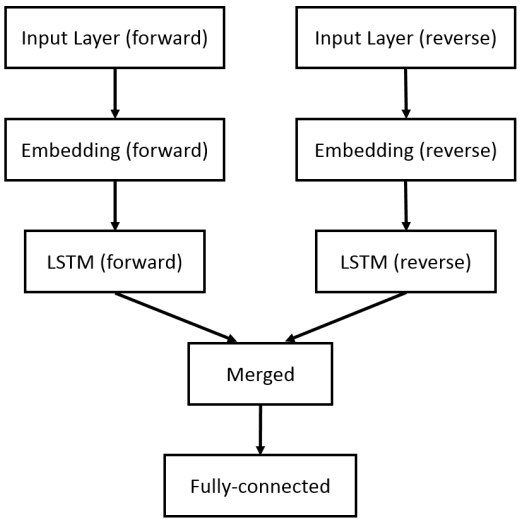

図 1 提案モデル

(注2)：http://www.dailymail.co.uk/

ングする．また，テストデータでは空所の前後に t 語以上の単語がある場合もあるが，モデルへの入力は前後 t 単語までであり，それ以上空所から離れている単語は無視される．

これを入力として得られるモデルの出力のベクトルと学習データに含まれる各単語のベクトルとのコサイン類似度を用いて性能評価を行う．性能評価については 4.3 節で述べる．

4. 評価実験

4.1 実験の設定

本実験では空所の前後何単語を入力として用いるかを表す t を 5 とした．すなわち各学習データは 11 語の単語列となる．これは 4.4.1 項で述べるパラメータ検証に基づいて決定した．

embedding 層の重みの初期値に利用する fastText のハイパーパラメータは `minCount` を 0 に設定した以外はデフォルトの設定を利用した．なお，fastText によって生成されるベクトルの次元数は 100 である．fastText の学習には提案手法のモデルと同じ学習データを用いる．実験で用いるデータについては 4.2 節で述べる．また，LSTM 層の出力は 128 次元とした．その次のマージ層での処理は入力された 2 つのベクトルの加算に設定した．すなわちマージ層への入力のうち，順方向の LSTM 層から出力されるベクトルの i 番目の成分を $u_{\mathrm{f},i}$，順方向の LSTM 層からの出力されるベクトルの i 番目の成分を $u_{\mathrm{r},i}$，マージ層の出力ベクトルの i 番目の成分を z_i とすると $z_i = u_{\mathrm{f},i} + u_{\mathrm{r},i}$ となる．

本モデルでは学習回数を 100 回とした．学習データの 9 割をモデルの学習に，残り 1 割を検証データとして 100 回学習を行い，そのうち検証データでの損失が最も少ないモデルをテストで用いる．

4.2 実験データ

学習データには text8 コーパス (注3) を用いる．text8 コーパスは，Mahoney によって 2006 年 3 月 3 日時点での英語版 Wikipedia (注4) の記事データを整形して作られた英文コーパスであり，Wikipedia の記事データの xml ファイルに対して以下のような前処理が施されている．

（1） 本文テキストと画像キャプション，アンカーテキスト以外の除去
（2） 大文字の小文字への変換
（3） 数字を 1 字ずつ英単語のつづりに変換
（4） a-z 以外の文字を全て半角スペースに変換
（5） 半角スペースの重複の除去
（6） 先頭 100MB のみ保持

text8 コーパスには 253,855 種類の英単語が含まれており，100MB の大きさのデータである．このコーパスから $2t + 1$ 語すなわち 11 語の単語列 5,667,039 件を抽出し，3.2 節で述べたように抽出した各単語列の前半 5 語と後半 5 語をモデルへの入力に，中央の 1 語をその入力に対する出力の教師ベクトルに用いる．

（注3）：http://mattmahoney.net/dc/textdata
（注4）：https://en.wikipedia.org/wiki/Main_Page

テストデータには 2000 年度から 2016 年度に行われた大学入試センター試験の本試験および追試験の大問 2A を用いる．text8 コーパスと同様の前処理を行ったのち，空所の選択肢が全て単語 1 語であり，かつ空所の前後に 2 語以上ずつ単語がある問題 158 問をテストに使用する．

4.3 性能評価

モデルの性能は以下の二つの観点で評価し，それぞれ実験 1，実験 2 と呼ぶ．また，評価尺度にはモデルが選択した語と正答の語の一致率，すなわち正解率を用いる．

実験 1　センター試験の選択肢四つから最も適切な単語を選択できるか．
実験 2　選択肢なしで空所に入る語を予測できるか．

本稿で用いるモデルは入力から空所に入る単語のベクトルを出力する．実験 1 ではこのベクトルと選択肢の 4 語の各ベクトルとのコサイン類似度を算出し，最も類似度の高い語を空所に入る語として決定する．実験 2 ではモデルの出力するベクトルと学習データに含まれる全単語とのコサイン類似度を算出し，最も類似度の高い語を空所に入る語として決定する．

実験結果を表 1 に示す．実験 1 は選択肢 4 語の中から 1 語を選び，実験 2 は 253,855 語の中から 1 語を選ぶ形式である．実験 1 では無作為に選んだ場合に比べて少しいい程度の結果となり，実験 2 では 1 問も正解することができなかった．

4.4 考察

4.4.1 提案モデルの構造とパラメータ

本実験で用いたモデルは 3.1 節で述べた構造であり，4.1 節で述べたパラメータを利用しているが，以下の構造やパラメータについても検証し，その中での最良の結果である．検証するモデルの構造およびパラメータを表 2 に示す．表 2 中の “マージ層での処理” の項目にある “連結” はマージ層への入力される 2 つベクトルを連結して出力する処理であり，出力されるベクトルの次元数は入力ベクトルの 2 倍となる．表 2 の上から順にモデルの設定を変更して学習を行い，テストデータに対する正解率が最も高い設定をそれ以降で用いる．また，初期設定は各項目の最も左にあるものを使用する．具体的には，まず t の値を 10, 5, 3, 1 と変えてそれぞれ学習を行う．このとき LSTM 層の種類は通常の LSTM，マージ層での処理は加算，全結合層は

表 1　モデルによる問題の解答結果

	正解の問題数	不正解の問題数	正解率 (%)
実験 1	51	107	32.2
実験 2	0	158	0

表 2　検証するモデルの設定

構造やパラメータ名	検証する構造の種類やパラメータの値
t	10, 5, 3, 1
LSTM 層の種類	通常の LSTM，双方向 LSTM
マージ層での処理	加算，乗算，連結
全結合層の配置	マージ層の後に 1 層，マージ層の後に 2 層， LSTM 層の後に 1 層ずつとマージ層の後に 1 層， LSTM 層の後に 1 層ずつとマージ層の後に 2 層

表 3　検証した各モデルの設定での結果

t	LSTM 層の種類	マージ層での処理	全結合層の配置	正解の問題数	不正解数の問題数	正解率 (%)
10	通常の LSTM	加算	マージ層の後に 1 層	28	68	29.1
5	通常の LSTM	加算	マージ層の後に 1 層	**51**	**107**	**32.2**
3	通常の LSTM	加算	マージ層の後に 1 層	45	116	27.9
1	通常の LSTM	加算	マージ層の後に 1 層	34	127	21.1
5	双方向 LSTM	加算	マージ層の後に 1 層	42	116	26.5
5	通常の LSTM	乗算	マージ層の後に 1 層	37	121	23.4
5	通常の LSTM	連結	マージ層の後に 1 層	33	125	20.8
5	通常の LSTM	加算	マージ層の後に 2 層	50	108	31.6
5	通常の LSTM	加算	LSTM 層の後に 1 層ずつとマージ層の後に 1 層	32	126	20.2
5	通常の LSTM	加算	LSTM 層の後に 1 層ずつとマージ層の後に 2 層	43	115	27.2

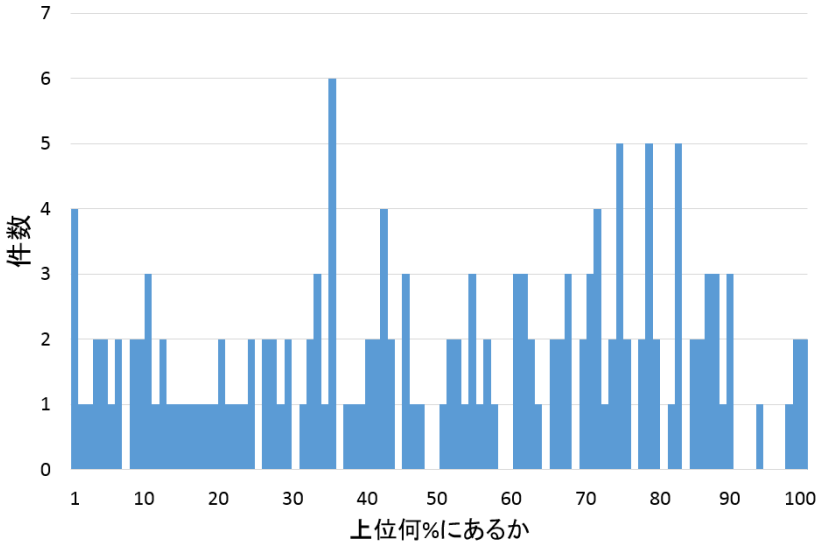

図 2　正解の語の順位の分布

マージ層の後に 1 層に固定する．テストデータでの正解率が最も高くなる t の値に固定して，続いて LSTM 層の種類を双方向 LSTM に変更して同様に学習を行い，テストデータでの正解率が高い方の種類の LSTM を用いる，といったように設定の検証を行う．これにより 10 通りの設定の中から正解率が最も高くなった設定が 3.1 節および，4.1 節に示す内容である．

検証したモデルの設定での結果を表 3 に示す．どの設定においても実験 2 の正解数が 0 だったため，実験 1 の結果のみ記載している．表 3 中に太字で示している最も正解率の高い設定での結果が 4.3 節の表 1 である．最良の正解率は 32.2%であるものの，他の設定を含めても選択肢から無作為に選んだ場合とあまり変わらない結果であった．

4.4.2　正解語の順位の分布

4.3 節の実験 2 では本モデルの出力するベクトルと学習データに含まれる全単語のベクトルのコサイン類似度を計算するが，この類似度の高い順に順位を割り当て，空所に入る正解の語の順位の分布を確認する．テストデータ 158 問それぞれの正解の語の順位が全単語中の上位何%にあたるかの分布のヒストグラムを図 2 に示す．正解の語の予測が適切ならば，図 2 のヒストグラムは左ほど高くなるはずであるが，実際の結果はそうなっていない．すなわち本モデルでは正解の語のベクトルと類似度の高いベクトルを予測できていないと言える．4.4.3 項で本モデルが上位に予測する単語について確認する．

4.4.3　上位の順位に予測される単語

本モデルによってテストデータ 158 問それぞれに対して空所に予測される単語の順位の上位 1%の語を確認すると，空所の前後の単語列が異なるにも関わらず各順位の上位 1%に予測される単語には重複が見られた．その出現回数の上位 10 件を表 4 に示す．表 4 より，例えば “ladislaus” はテストデータ 158 問中 157 問で順位の上位 1%に含まれていることが分かる．そこで，各テストデータに対して本モデルが出力するベクトル間のコサ

表 4　モデルが予測した各順位の上位 1%に重複して現れる単語

単語	出現回数	単語の品詞
ladislaus	157	名詞 (人名)
habriot	157	名詞 (人名)
wladislaus	157	名詞 (人名)
fahkr	157	名詞 (人名)
ausfrid	156	名詞 (人名)
radulf	156	名詞 (人名)
spartacist	156	名詞 (人種名)
rkisyrj	156	名詞 (地名)
yorkist	156	名詞 (人種名)
bogislaus	156	名詞 (人名)

イン類似度を算出したところ，その平均値は 0.867810，中央値は 0.888832 となった．すなわち本モデルから出力されるベクトルは入力に関わらず似たベクトルになっている．

また，表 4 では人名などの固有名詞のみが現れている．学習データでの空所にあたる語，すなわち 11 語の単語列の中央の 1 語や，テストデータでの空所に入る正解の語について，出現回数の多い単語の上位 10 件を表 5，表 6 に示す．学習データやテストデータでも現れる語に偏りはあるものの，表 4 中の語のような固有名詞には偏っていないことから，本モデルによって適切な出力が得られない原因は学習データやテストデータではなくモデル自体だと考えられる．

4.4.1 項から 4.4.3 項で述べたように，本モデルでは空所の前後の単語列から空所に入る単語の適切なベクトルを出力できておらず，モデルの構造やパラメータの検証を行っても正解率に大きな変化は得られなかった．この原因には本実験で検証したモデルの構造では学習が十分にできないことやデータの入出力の方法が不適切であることが考えられる．本モデルは空所の前後 2 つの単語列からその間の 1 単語を予測するモデルであり，空所の前と後を別の入力として与えているが，これらを連結して 1 つの単語列を入力とする方法や，出力についても空所の 1 単語ではなく，空所を補充した後の単語列を出力する方法が考えられる．また，近年注目されている attention メカニズムを本手法では用いていないため，モデルの入出力の形式やモデルの構造自体についても見直すことが今後の課題である．

表 5　学習データ 5,667,039 件中で空所に入る単語の出現回数

単語	出現回数	単語の品詞
the	353793	前置詞
of	197699	前置詞
and	138694	接続詞
one	136575	数詞
in	123776	前置詞
a	108892	冠詞
to	105415	前置詞
zero	88279	数詞
nine	84062	数詞
two	64477	数詞

表 6　テストデータ 158 件中で空所に入る単語の出現回数

単語	出現回数	単語の品詞
in	4	前置詞
make	3	動詞
which	3	関係詞
on	2	前置詞
for	2	前置詞
down	2	副詞
during	2	前置詞
out	2	副詞
until	2	前置詞
happen	2	動詞

5. ま と め

本稿では NN を用いた英文空所補充問題の解法として，空所の前後それぞれの単語列を LSTM 層への入力とし，2 つの LSTM 層の出力をマージする NN のモデルを作成し，空所に入る 1 単語の予測を行った．text8 コーパスを学習データ，大学試験センター試験の問題をテストデータとして実験を行ったところ，センター試験の選択肢 4 語の中から空所に入る語を予測する際の正答率は 32.2%であり，学習データに含まれる全単語からの予測では 1 問も正解することができなかった．提案モデルでは入力の違いに関わらず似たベクトルを出力しているため，学習時の入出力データの形式やモデルの構造そのものを見直すことが今後の課題である．

文　献

[1] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant. The CoNLL-2014 shared task on grammatical error correction. Proceedings of Conference on Computational Natural Language Learning: Shared Task (CoNLL), 2014

[2] M. Junczys-Dowmunt and R. Grundkiewicz. Phrasebased machine translation is state-of-the-art for automatic grammatical error correction. Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 2016.

[3] Z. Xie，A. Avati，N. Arivazhagan，D. Jurafsky，and A. Y. Ng. Neural language correction with character-based attention. arXiv:1603.09727，2016.

[4] 小山田創哲，兼村厚範，石井信，“根拠を明示するニューラル文法誤り訂正”，DEIM Forum 2017，G4-3，2017.

[5] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2-3):103-134, 2000.

[6] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom. Enriching word vectors with subword information. arXiv:1506.03340，2015.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, pp. 3111-3119, 2013.

[8] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, pp. 1-12, 2013.

[9] P. Bojanowski，E. Grave，A. Joulin and T. Mikolov 2016. Enriching word vectors with subword information. arXiv:1607.04606，2016.

[10] A. Joulin，E. Grave，P. Bojanowski and T. Mikolov 2016. Bag of Tricks for Efficient Text Classification. arXiv:1607.01759，2016.