The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

擬似誤りコーパスを用いた天気予報原稿のニューラル誤り検出

Neural Error Detection for Weather Forecast Manuscript by Pseudo Error Corpus

白井稔久 *1　萩行正嗣 *2　小町守 *1
Naruhisa Shirai　Masatsugu Hangyo　Mamoru Komachi

*1首都大学東京　*2株式会社ウェザーニューズ
Tokyo Metropolitan Univercity　WEATHERNEWS INC.

In this paper, we propose a neural method for detecting errors in Japanese weather forecast manuscripts. First, we analyze errors in weather forecasts to understand how native Japanese mistake. According to our analysis, we found native Japanese tend to cause errors in three types: particle error, conversion error and typo. in this paper, we focus on particle and conversion errors. However, any corpora written by native Japanese are not large enough for supervised learning. Therefore we use pseudo error corpus to augment training data. We generate pseudo particle errors by confusion matrix and we also generate pseudo conversion errors by back transliteration. As a result, we find pseudo corpus is effective for neural error detection for text written by native Japanese.

1. はじめに

天気予報原稿は一般的に人手で記述されているため，誤りを含んでいる場合がある．それらの誤りは公開する前に校正する必要がある．通常これらの誤りは人手での多重チェックなどで公開前に校正されているが，この校正には大きなコストがかかっている．

日本語母語話者が記述したテキストの自動誤り検出を行うには，テキストに誤りがアノテーションされているコーパスが少ないため，教師あり学習の手法をとることは困難である．また，日本語学習者が記述したテキストに比べて誤りが少なく，加えて誤り傾向が異なる可能性があり学習者のコーパスを学習にそのまま使用することは不適切であると考える．RNN 言語モデルを用いた誤りの自動検出 [9] では，誤警報率が非常に高く日本語母語話者の誤り検出が難しいことが報告されている．

そこで我々はまず誤り傾向を分析するためウェザーニューズ *1 の天気予報原稿コーパスを用いた．このコーパスは 2 年分の天気予報原稿があり，編集前の原稿と編集後の原稿がペアになっているため，擬似的に校正コーパスとみなすことができる．実際に含まれていた編集前後の文章対を図 1 に示す．このコーパスを用いた分析の結果，誤りの大部分は誤変換，助詞誤り，タイポの 3 つに分類できることが分かった．

そのため我々は教師あり学習が適用可能な誤変換と助詞誤りに着目しそれらの検出を行うことにした．しかし，それらを教師ありの手法を用いて検出するにはアノテーションされているデータ数が少ない．そこで我々は小規模データでの教師あり学習に有用であることが分かっている擬似コーパス [4] でコーパスを拡充するために，擬似誤りを生成した．

擬似誤りの生成は助詞の擬似誤りと誤変換の擬似誤りで分けて生成した．助詞の擬似誤りに関しては，助詞ごとに誤り傾向を分析し，その傾向に従って分布を作り擬似的な誤りを生成した．誤変換に関しては分析したコーパス全体で単語単位で誤変換をしている割合を基に，単語を別の同音の単語に変換して擬似誤りを生成した．

本研究の貢献は以下の 2 つである．

- 天気予報原稿の誤り検出のためのコーパスを分析した
- 擬似誤り生成によるコーパスの拡充が日本語母語話者が記述したテキストのニューラル誤り検出に関して有用であることを示した

誤りが含まれる文章対
編集前
今日は雲優勢のスッキリしない空。
髪が乱れるほどの風が強いのでご注意下さい。
日差しが少なくてもムシ暑くなります。
編集後
今日は雲優勢のスッキリしない空。
髪が乱れるほど風が強いのでご注意下さい。
日差しが少なくてもムシ暑くなります。

誤りが含まれない文章対
編集前
今日も雨が降り続きます。
激しく降る可能性があるので大きめの傘やレインコート・ブーツが良さそうです。
河川の増水・道路冠水・土砂災害にご注意下さい。
編集後
今夜～今日も雨が降り続きます。
激しく降る可能性があるので大きめの傘やレインコート・ブーツが良さそうです。
河川の増水・道路冠水・土砂災害にご注意下さい。

図 1: 編集前後の文章対の例．下線部が編集された部分である．

2. 関連研究

日本語母語話者が記述したテキストの誤り訂正の研究として，新納らは平仮名 n-gram を用いて誤りを検出し，訂正する手法を提案した [5]．また，南保らは文節内の特徴からルールを自動作成し，ルールベースで，日本語の助詞誤りを検出し，校正する手法を提案した [8]．これら 2 つの手法は我々と同じく，日本語母語話者が記述したテキストを対象にしていて，特に南保らの研究とは助詞誤りに着目した点で我々の研究と共通する．一方我々の研究では教師あり学習を用いている．

また，日本語学習者が記述したテキストに対する自動訂正の研究も広く進められている．今村らは日本語学習者が助詞を間違えやすいことを指摘し，その助詞を，間違えやすい助詞の単語テーブルを用いることによって修正する手法を提案した [4]．また，今村らは小規模の誤りデータから擬似誤りを生成し，コーパスを拡充した．この研究は助詞に着目した点，擬似誤りを生成した点で我々と共通するが，我々の提案手法では助詞だけではなく，誤変換も対象としていて，誤りの検出のみで

連絡先: 白井 稔久，shirai-naruhisa@ed.tmu.ac.jp
*1　https://weathernews.jp

訂正はしない．また，今村らは訂正に CRF を用いているが，本研究では Bi-LSTM を用いた RNN を使った．

水本らは，語学学習 SNS である Lang-8 から添削ログを抽出しコーパスを作成し，そのコーパスを用い，文字単位での修正と，文字-単語間での修正の二つの手法を提案した [6]．彼らが提案した手法は，統計的機械翻訳モデルを用いて誤りを修正するものである．我々の研究とコーパスを作成した点で共通するが，我々の提案手法では擬似誤りを生成してコーパスの拡充を図っている．また，本研究では RNN を用いている．

3. 擬似誤りコーパスの作成

我々は教師あり学習を行うには少ないコーパスを拡充するために，擬似誤りコーパスを作成した．擬似誤りコーパス内に含まれる擬似誤りは，助詞誤りと誤変換で異なる方法で作成した．擬似的な助詞誤りは実際に天気予報原稿コーパスで誤った助詞を基に誤りを作成した．擬似的な誤変換は天気予報原稿コーパスに含まれる誤変換の割合で元の単語を誤変換させて作成した．また，それらの誤変換は元の単語を平仮名にした後に再変換し，元の単語と異なるものに置換することで作成した．

3.1 天気予報原稿の誤り傾向分析

誤りの傾向を分析するために本研究ではウェザーニューズ社の天気予報原稿コーパスの分析を行った．このコーパスには 2014 年と 2015 年の 2 年分の天気予報原稿の，編集前の文章と編集後の文章対が入力されている．それらの文章対の総数は 100,931 対である．

これらの文章対における編集の内容は，文法誤りなどの校正だけでなく，原稿内容や表現を大きく書き換えるようなものも含まれる．それらを除外するために，編集前後の文章間の文字単位での編集距離が 1 以上 5 以下の文章対を抽出した．本研究で学習データとして用いた 2014 年のデータを対象に抽出された文章対は 2,575 対で，文対数は 7,765 文対だった．また，天気予報原稿内に 5 文字以下の文が含まれている可能性は非常に低いと考え，編集距離で抽出した文章対に編集前後で文数が異なるものは含まれていないと判断した．

その後，日本語形態素解析システム JUMAN7.0[*2] を用いて文を形態素解析し，編集距離を用いた動的計画法で形態素単位でアライメントをとり，異なり形態素対およびその異なりが含まれる文を人手で確認し，実際に誤りであると判断したものの編集前後の形態素対を記録し分析した．その結果誤りは大きく分けて誤変換，助詞誤り，タイポの 3 つに分類できることが分かった．

3.2 誤りタグ付きコーパスの作成

3.1 項と同様にして誤りであると判断した単語対にアノテーションを付与し，学習データとしてコーパスを作成した．誤りにアノテーション付与する際，以下のルールに基づきアノテーションを付与した．下記のルールに基づきアノテーションを付与する例を図 2 に示す．

置換 編集前の単語と編集後の単語を置換して文が成立する場合，その単語に誤りタグを付与

文の不成立 編集前の文が成立していない場合に原因と思われる単語に誤りタグを付与

余字 編集前の単語を削除すると文が成立する場合その単語に誤りタグを付与

*2 http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

置換
編集前：こまめに 水分 を 取って 熱中 症 対策 を 万全に して ください。
編集後：こまめに 水分 を 摂って 熱中 症 対策 を 万全に して ください。

文の不成立
編集前：室内 でも 熱中 症 に なること が あり 体調 管理 は 万全に。
編集後：室内 でも 熱中 症 に なること が ある 体調 管理 は 万全に。

余字
編集前：今日 の 今日 の 朝 は 雨 の 可能性 が ありますが ～
編集後：今日 の 朝 朝 朝 は 雨 の 可能性 が ありますが ～

脱字
編集前：今日 は 夏 空 広がり ます が 急な 雨 も あり ます。
編集後：今日 は 夏 空 が ます が 急な 雨 も あり ます。

図 2: ルールに基づいてタグを付与した例．下線部の単語に誤りのタグを付与する．編集後の文は編集前の文中における単語にアライメントを取っている単語の列であるため，実際の文ではない．また，編集前の文中における 1 単語に対して編集後の文中における複数の単語がアライメントを取っている場合，最も文頭に近い単語のみを出力している．

表 1: 学習・開発・評価データの詳細

	学習	開発	評価
誤変換	90	5	2
助詞誤り	76	5	6
タイポ	190	8	20
誤りの総数	356	18	28
総文数	7,765 文	3,842 文	2,971 文

脱字 編集前の文に明らかに単語が不足している場合，不足していると思われる位置の直後の単語に誤りタグを付与

上記に該当しない単語は誤っていないものとして誤りタグを付与しない

また，上記とほぼ同様の手順で誤りがアノテーションされているテストデータも作成した．相違点は，2014 年のデータではなく 2015 年のデータを対象にしたこと，編集距離が 1 以上 5 以下の文章対ではなく 0 以上 5 以下の文章対を用いたこと，そうして抽出された文章対の中から，季節や時期による文章の内容の偏りを防ぐために各月ごとに 200 対ずつランダムサンプリングしたことである．その結果 2,400 文章対からなる 6,813 文のアノテーション付きのデータを作成した．さらに作成したコーパスを各月から 100 対ずつ，合計 1,200 対ずつ開発データと評価データに分割した．学習データ，開発データ，評価データ誤りの種類毎の件数，総文数を表 1 に示す．

3.3 擬似誤りを含む文の生成

本研究では今回作成したコーパスが小規模であることから，小規模のコーパスでも誤りが検出できるよう擬似的に誤りを作成した．本研究では誤変換と助詞誤りを対象に擬似誤りを生成し，学習データの拡充を行う．

学習者が記述したテキストの擬似的な誤りの生成は一般的に実際に誤ったような誤りの生成割合で行われる [4]．しかし，予備実験によって本コーパスにおいてはこの手法はあまり有用でないことが分かった．そこで，本研究では 3.2 項で作成したコーパスの誤った単語対を基に，一様分布を用いて抽出前の全ての編集後の文章を誤らせることで擬似的に誤りを生成した．

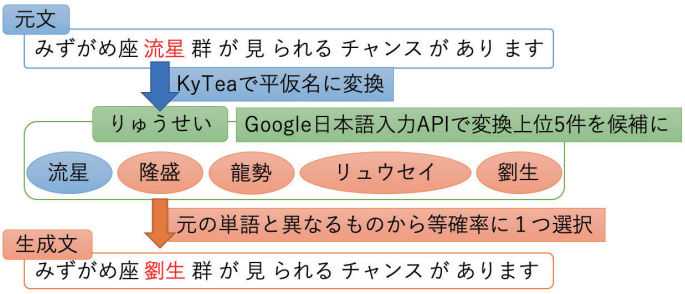

図 3: 擬似誤変換生成の例

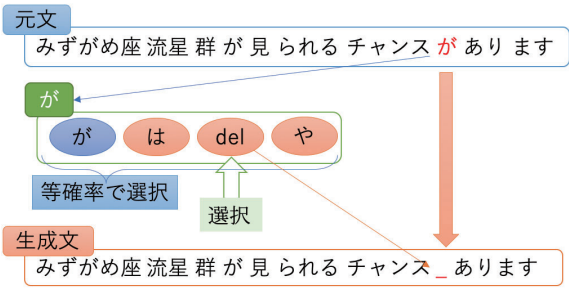

図 4: 擬似助詞誤り生成の例

我々は誤変換はどのような単語に対しても起こりうると考えた．そこで，作成したコーパスの全単語を定めた割合に基づき再変換し，擬似的な誤変換を生成した．誤変換の作成は 1 度単語を京都テキスト解析ツールキット KyTea [2] を用いてかなに直し，その後 Google 日本語入力 API [*3] を用いて，ランキング上位 5 件から無作為に元の単語と異なる単語を選び，擬似的な誤変換を作成した．擬似誤変換の生成の例を図 3 に示す．

助詞の誤りに関しては，2 つの方法で擬似誤りを生成した．1 つは 3.2 項で作成した学習コーパス中で出現した編集後の各助詞 w について編集前では誤っている各単語 $w_i = \{w_1, w_2, ...w_L\}$ と元の単語に対して一様に確率を割り当てる一様分布に従い擬似的な助詞の誤りを生成した．また，元の単語を選んだ場合誤りタグは付与しない．助詞の擬似誤り生成の例を図 4 に示す．もう 1 つは擬似誤変換の生成と同様に一定の割合で助詞誤りを発生させる方法である．全ての助詞に対して一定の割合に基づきその助詞を誤らせるか決め，誤らせる場合学習コーパス内で出現した誤り方から無作為に 1 つ選択肢誤らせる．また，学習コーパス内で誤らなかった助詞を誤らせることが選択された場合，その助詞を削除することで擬似助詞誤りを生成した．

4. 助詞誤りと誤変換の検出実験

4.1 誤り検出器

本研究では Bi-LSTM [1] を用いた誤り検出器で実験を行う．

入力文 $S = (w_1, w_2, \cdots, w_n)$ の各単語 w_t は単語ベクトル $e_t \in \mathbb{R}^{de \times 1}$ に変換される．n は文長であり，de は単語ベクトルの次元である．単語ベクトルから LSTM により順方向の隠れ層 $\overrightarrow{h_t} \in \mathbb{R}^{dh \times 1}$ と逆方向の隠れ層 $\overleftarrow{h_t} \in \mathbb{R}^{dh \times 1}$ を作成する．dh は隠れ層の次元とする．$\overrightarrow{h_t}$ と $\overleftarrow{h_t}$ を連結することで最終的な隠れ層 $h_t^{(lstm)} \in \mathbb{R}^{2dh \times 1}$ を獲得する．隠れ層 $h_t^{(lstm)}$ を以下のように線形変換しソフトマックス関数を使い正誤タグの確率分布 $p_t \in \mathbb{R}^{t \times 1}$ を獲得する．t はタグのサイズであり，正誤のどちらかのタグを予測するためサイズは 2 である．

$$p_t = \mathrm{softmax}(W_h h_t^{(lstm)} + b_h) \quad (1)$$

$W_h \in \mathbb{R}^{v \times dh}$ 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．

誤差関数である $loss$ は交差エントロピーによって以下の式を用いて計算される．

$$loss = -\sum y_t \log p_t \quad (2)$$

y_t は正解のタグであり学習データ内で正誤のどちらかが付与されている．

*3 https://www.google.com/inputtools/try/

表 2: 実験に用いた各学習データの文数．「orig」は 3.2 項で作成したコーパス，「pp」は助詞の擬似誤りを生成したコーパス，「conv」は誤変換を擬似生成したコーパスを示す．

学習データ	文数
orig	7,765 文
orig+pp	115,744 文
orig+conv	115,744 文
orig+pp+conv	225,305 文

4.2 実験設定

入力はあらかじめ学習済みの朝日新聞単語ベクトル [7] を用いてベクトル化した．このときの埋め込み層は 300 次元である．このモデルは PyTorch 1.0 で実装し，出力層の値を基に誤っている確率を出力する．この確率が 0.5 を超えているものを誤りとして検出する．隠れ層は開発データを用いて 1 層で 1024 次元に定めた．また，出力層は 200 次元，初期化は −0.1 から 0.1 の間でランダムに初期化した．バッチサイズは 64，最適化には ADADELTA [3] を用いて学習した．

学習には 3.2 項で作成したコーパスとそれに擬似誤りを生成したコーパスを用いる．また，今回の誤り検出では擬似誤りを加えたことによる有用性を確かめるために助詞誤りと誤変換のみを誤りとして検出する．よって学習データのタイポの誤りは誤りでないものとして学習した．開発データを用いて各エポックで再現率が最大のときに適合率が最も高いエポックのものを評価データの実験に用いた．また各データの文量を表 2 に示す．

評価には適合率と再現率を用いた．適合率はシステムが誤りだと判断した単語の内，実際に誤りである割合である．再現率はコーパス内の検出の対象である全ての誤りである単語の内，システムが検出した誤りの割合である．

4.3 実験結果

実験結果を表 3 に示す．表を見ると擬似誤りコーパスを学習データとして追加すると適合率，再現率ともに上昇していることがわかる．ただ，再現率は全体的に低く，誤りをほとんど検出できていないことがわかる．また，3.2 項で作成したコーパスだけでは学習のための文数が非常に少ないため誤りを検出することができていないことが分かる．

また，コーパス内で擬似誤りを生成する割合を変えて実験した結果を表 4 に示す．この実験のデータ量は orig+conv，orig+pp と同一である．表を見ると擬似誤りを生成する割合を増やすと性能が向上していることが分かる．特に助詞誤りは生のが著しく向上している．また，予備実験で行った助詞の擬似誤りを実際に誤った分布に従って生成する実験で擬似誤りを生成する割合を増やしたが，特に性能の向上が見られなかった．

表 3: 誤り検出実験の結果．pp*は元の誤り分布に従い助詞誤りを生成したコーパスで学習したものである．@以下の数字は擬似誤りの生成割合である．

学習データ	分割	適合率	再現率
orig	開発	0.00	0.00
	評価	0.00	0.00
orig+pp*	開発	0.00	00.0
	評価	0.00	00.0
orig+pp	開発	0.21	20.0
	評価	0.26	25.0
orig+conv	開発	1.75	20.0
	評価	0.00	0.00
orig+pp*+conv	開発	0.16	20.0
	評価	0.00	0.00
orig+pp+conv	開発	1.38	50.0
	評価	0.29	37.5
orig+pp@50.0%+conv@50%	開発	1.03	70.0
	評価	0.71	50.0

表 4: 擬似誤りの生成割合を変えた実験結果

		orig+conv		orig+pp	
生成割合	分割	適合率	再現率	適合率	再現率
0.01%	開発	0.00	0.00	0.00	0.00
	評価	0.00	0.00	0.00	0.00
0.1%	開発	3.30	30.0	0.00	0.00
	評価	0.00	0.00	0.00	0.00
1.0%	開発	0.79	20.0	13.3	20.0
	評価	0.00	0.00	16.6	12.5
10.0%	開発	0.90	30.0	8.33	10.0
	評価	0.31	12.5	7.14	12.5
20.0%	開発	0.71	30.0	3.03	30.0
	評価	0.52	25.0	3.09	37.5
30.0%	開発	0.69	30.0	0.77	40.0
	評価	0.49	25.0	1.13	62.5
40.0%	開発	0.74	30.0	1.26	50.0
	評価	0.51	25.0	1.47	62.5
50.0%	開発	1.50	30.0	1.24	50.0
	評価	1.03	25.0	1.52	62.5

5. 分析

助詞誤りに関しては生成割合を一様分布に変えた結果，適合率は非常に低いが再現率は上がり，一定の生成割合で助詞誤りを生成した結果，適合率と再現率両方の上昇が見られた．このことから助詞の誤り検出に関しては母語話者の誤る割合よりも大きい割合で擬似誤りを生成した方が，性能が向上することが分かった．又，一定の割合での助詞誤りを生成したコーパスで学習したモデルは，“ は ”や“ が ”の助詞誤りは検出できる傾向にあった．これは“ は ”と“ が ”は元のデータ内でも誤り件数と誤り方の種類も多いため，ほとんどの誤り方を再現できたからではないかと考える．

誤変換に関しては生成割合を上昇させた結果，評価データ内の誤変換は 2 件とも検出できているが開発データ内の誤変換は 2 件検出できていない．これは両方とも“ うだるような ”が“ うだる様な ”に変換されてしまっているものだった．この変換を学習データ内には存在したが単語分割が開発データでは“ よう_な ”と分割されていたのに対し，学習データでは“ ような ”と分割されていたため，実質学習データ内に存在しない誤りになってしまったためだと考える．

助詞誤りの誤検出に関しては“ は ”，“ の ”，“ を ”の誤検出が多く見られた．この 3 つの助詞は全ての擬似助詞誤り生成法で誤りとして生成する割合が他の助詞よりも高いため，誤りとして誤検出してしまう傾向にあるのではないかと考える．

誤変換の誤検出に関しては“ 熱 ”という単語をよく誤検出してしまう傾向にあった．これは学習データ内の“ 熱 ”という単語ほとんどに誤りのアノテーションが付与されており，出現してしまうこと自体が誤りだと認識したためだと考える．

6. おわりに

本研究では天気予報原稿の誤り検出において教師あり学習を行うためのコーパスの作成と，そのコーパスを拡充するために実際に日本語母語話者が誤った確率を基に擬似誤りの生成を行なった．擬似誤りを生成する提案手法は結果として日本語母語話者が記述した日本語テキストのニューラル誤り検出に有用であることが分かった．加えて実際の誤る割合，および分布に従って擬似誤りを生成するよりも擬似誤りの生成割合を上げる方がモデルの性能が向上することが分かった．しかし，Bi-LSTM を用いたモデルでの検出は未知の誤りの検出が非常に困難であることが分かった．また，本研究の提案手法では適合率が低いため，適合率を上げる手法を検討する必要がある．

参考文献

[1] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, Vol. 18, pp. 602–10, 07 2005.

[2] Graham Neubig and Shinsuke Mori. Word-based partial annotation for efficient corpus construction. *LREC*, pp. 2723–2727, 2010.

[3] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, Vol. abs/1212.5701, , 2012.

[4] 今村賢治, 齋藤邦子, 貞光九月, 西川仁. 小規模誤りデータからの日本語学習者作文の助詞誤り訂正. 言語処理学会論文誌, Vol. 19, No. 5, pp. 381–400, 2012.

[5] 新納浩幸. 平仮名 n-gram による平仮名文字列の誤り検出とその修正. 情報処理学会論文誌, Vol. 40, No. 6, pp. 2690–2698, 1999.

[6] 水本智也, 小町守, 永田昌明, 松本裕治. 日本語学習者の作文自動誤り訂正のための語学学習 SNS の添削ログからの知識獲得. 人工知能学会論文誌, Vol. 28, No. 5, pp. 420–432, 2013.

[7] 田口雄哉, 田森秀明, 人見雄太, 西鳥羽二郎, 菊田洸. 同義語を考慮した日本語単語分散表現の学習. 情報処理学会第 233 回自然言語処理研究会, Vol. 2017-NL-233, pp. 1–5, 2017.

[8] 南保亮太, 乙武北斗, 荒木健治. 文節内の特徴を用いた日本語助詞誤りの自動検出・校正. 情報処理学会第 181 回自然言語処理研究会, Vol. 2007, No. 94, pp. 107–112, 2007.

[9] 白井稔久. RNNLM を用いた日本語テキストの誤字・脱字検出および再変換を用いた誤変換検出. 首都大学東京卒業論文, 2018.