日本語学習者向けの文法誤り検出機能付き作文 用例検索システム

Example Retrieval System using Grammatical Error Detection for Japanese as a Second Language Learners

新井 美桜
Mio Arai

東京都立大学
Tokyo Metropolitan University
cherry.m@mac.com

金子 正弘
Masahiro Kaneko

(同 上)
kaneko-masahiro@ed.tmu.ac.jp

小町 守
Mamoru Komachi

(同 上)
komachi@tmu.ac.jp

keywords: grammatical error detection, example sentence retrieval system

Summary

Example sentences retrieval systems display examples that match the input query. They help language learner's writing. Learners can use the pattern of the examples in their writing. If the query word is correct, learners can find examples in a native corpus. However, if the query word is incorrect, it is impossible to find appropriate examples using ordinary search engines. Existing example retrieval systems do not include grammatically incorrect examples, or only present a few examples, if any. Even if a retrieval system has a wide coverage of incorrect examples along with the correct counterparts, learners need to know whether their query includes errors. Considering the usability of retrieving incorrect examples, our proposed method uses a large-scale learner corpus and presents correct expressions along with incorrect expressions using a grammatical error detection system so that the learner does not need to be aware of how to search for examples. Learners can recognize the wrong part in the input query and know how to revise the wrong part when they check correct expressions along with incorrect expressions. Evaluations indicate that our method improves the accuracy of example sentence retrieval and the quality of a learner's writing. To the best of our knowledge, our system is the first incorrect example sentence retrieval system using neural grammatical error detection.

1. はじめに

近年、言語学習者向けの作文支援システムが多く開発されている。その中には、学習者作文の文法誤りを自動で訂正するシステムや、学習者作文を自動評価し点数をつけるシステムなどがある。このような作文支援システムにより、学習者及び教育者の負担を減らすことができる。

本研究では、作文支援システムとして、用例検索システムの構築に取り組む。用例検索システムは、多数の用例を収録したコーパスから、文や語句などのユーザの検索クエリに応じて用例を提示するシステムである。言語学習者が作文時に書きたい内容を思いついていても、部分的な単語や表現などのみで適切な表現が思いつけない場合にこのシステムが役立つ。

一般的な用例検索システムにはいくつかの問題点がある。一つは、正しいクエリを入力しなければ適切な用例を提示できないという点である。学習者が誤用の

クエリを入力することは十分に考えられることであり、その場合は適切に検索することができない。この問題はシステムが十分な量の誤用例を持つことで解決できる。もう一つの問題は誤用例を検索できる場合のもので、学習者は提示された誤用例中の誤りの部分やその修正方法がわからないという点である。これは誤用例と共に正用例を提示することによって解決できる。李らは誤用例と正用例を持つコーパスの検索ツールを作成している[李 08]が、正用例と誤用例を同時に検索することができない。また、誤用例と正用例を同時に提示するシステム[Nishina 14]もあるが、用例の総数が少ない。更にもう一つ、誤用例と正用例を併せて提示可能な場合の問題点として、学習者自身が誤りを認識していない場合、どの部分を直すべきか分からず、混乱を招く可能性があることが挙げられる。この問題は学習者に誤り箇所を知らせることで払拭できる。

そこで、本研究では、誤用例を提示することの有用性に着目し、第二言語学習者のための文法誤り検出機能

図 1: 検索システムのユーザインタフェース。誤用検索された場合（左）と正用検索された場合（右）。

付き大規模誤用例文検索システムを構築した。提案システムは、文法誤り検出機能により入力文の誤りを自動で検出することで学習者に誤りを気づかせることができ、かつ、正用例付き誤用例を提示することで学習者自身がどのように自分の作文を訂正するかを判断できる検索システムとなっている。用例データには、学習者の書いた誤用文に母語話者の書いた添削文が付与されている大規模なデータセットを用いた。このデータセットを利用することで、学習者は自身の文がどのように間違っているか分かり、どのように直すかの判断の参考にできる。このシステムは、誤りが検出された場合は誤り箇所を、されなかった場合はユーザが入力した文をクエリとして全文検索を行い、入力クエリとの類似度が高い順に用例を表示する。

図 1 に提案する日本語学習者向けの用例検索システムのユーザインタフェースを示す。ユーザインタフェースは 2 種類あり、図 1 の左が検索クエリに文法誤りが含まれていて誤用検索された場合、図 1 の右が正しい検索クエリによって正用検索された場合である。以下に要素の説明を記す。

- ① **検索窓** ここに入力された検索クエリを検索する。検索クエリは文、フレーズ^{*1}または単語を想定する。
- ② **文法誤り検出結果** 誤り検出を行った結果が表示される。検出された誤り部分は赤く表示される。
- ③ **検索結果** 検索クエリに適合した検索結果が表示される。検索結果は訂正箇所が太字で強調された誤用例と正用例の文対である。クエリに誤りが含まれる場合、クエリが全て正しい場合共に、誤用例と正用例をペアで表示する。文法誤り検出の結果検索されたクエリが書かれた緑色のボタンをクリック

*1 ここでフレーズとは、「ニュージーランド」や「助けられる」などの 1 つまたは数個の形態素で構成される 1 つの意味を表す句のことを指すこととする。

クすると、そのクエリに適合する全ての用例が展開される。

日本語学習者にとって助詞は非常に誤りやすい項目であるため、使用例として、学習者であるユーザが助詞を誤った文をクエリとして入力した場合を示す。

- (1) ユーザが「日本語が勉強する。」と検索窓に入力する。
- (2) システムが誤り検出を行い、「が」が誤りだと検出し、図 1 の左の検索結果のように助詞「が」の誤りを含み、「日本語が勉強する。」に類似した用例を上位に提示する。
- (3) 各誤用例には添削後の正用例が付いているため、ユーザは「日本語を勉強する。」が正しい使い方であることを把握することができる。

本研究の主な貢献は以下のとおりである。

- 提案システムは文法誤り検出機能を持つ最初の用例検索システムである。この機能により、学習者はクエリのどの部分が間違っているのかを認識できる。
- 学習者の書いた誤り文である誤用例とそれに対応する添削文を対にして提示するため、学習者は用例を参考にして誤りを修正できる。
- 提案検索システムと比較検索システムを用いて評価実験を行い誤りタイプを分析した結果、提案検索システムは誤形成や脱落、余剰の誤りを文法誤りとして認識できることを示した。
- 提案システムを使用した被験者実験で言語学習者のライティングスコアが向上することを示した。

2. 関連研究

2.1 用例検索システム

最も一般的な用例検索システムは、Web ベースの検索エンジンである。しかしながら、Web ベースの検索エンジンは言語学習者のために例文を検索するわけではないため、誤用検出や文法誤り検出の機能はない。以下に主要な言語学習者向けの用例検索システムを記す。いずれも作文執筆を支援する機能を持つが、学習者を支援するための文法誤り検出や誤用検出の機能は持たない。

§1 日本語学習者向けの用例検索システム

李ら [李 08] は日本学習者の発話データを書き起こした資料に言語情報を付与した『タグ付き KY コーパス』を公開した。『タグ付き KY コーパス』は検索したい語を入力するとコーパスから学習者の正用例、誤用例を検索できる。しかし、『タグ付き KY コーパス』に含まれる誤用例は形態素数が 173,198 と非常に少なく、頻繁に使用するであろう語句にも誤用例が存在しない場合が多くある。また誤用例に対応する正用例が存在しないため、学習者が作文に一致する誤用例を見つけても、訂正の仕方が分からず、参考にすることが難しい。また、そもそも入力したクエリが誤用であると学習者自身が認識していないと誤用検出を行えないため、学習者の利用が困難であると考えられる。

東京外国語大学のコーパスに基づく言語学教育研究拠点は『国際日本語学習者作文コーパス及び誤用辞典*2』を構築した。このシステムは検索したい学習者の母語や年齢、性別などを選択するとその条件に合った文を検索ワードとともに検索でき、検索したキーワードに一致した文が KWIC (keyword in context) 形式で表示される。この検索システムは教育者を主な対象にしたもので、日本語教育者はこの検索システムを使うことで、学習者の誤りの特徴を得ることができるが、学習者が扱いやすいような用例の表示の仕方はしていない。そして、誤用検出を行い誤り箇所とそれに対する訂正文を表示する機能もあるが、総文数が非常に少ないため、頻出単語の誤用であっても誤用例が存在しない場合が多い。また、文法誤り検出機能を持たないため学習者は自分の検索ワードに誤りが含まれているかどうかを認識できず、どのように検索すべきか分からない。

本研究では、学習者の書いた誤用文に母語話者の書いた添削文が付与されているコーパスを用いることで、添削文付きの誤用例検索を可能にし、かつ、誤り検出機能を追加し、学習者が自身の文に含まれる誤りを認識していない場合における誤用検出も可能にした。

§2 英語学習者向けの用例検索システム

FLOW [Chen 12] は、英語学習者が母語で文を書く際に、言い換えを利用してリアルタイムに英単語の候補

を提示するシステムである。このシステムとは異なり、我々のシステムは訂正済みの学習者コーパスに基づいて、学習者が書いた誤りを含む文とそれに対応する訂正文を対にして提示する。

WriteAhead [Yen 15] は英語学習者が英語の文を書く際にリアルタイムで用例を提示する。このシステムは、コロケーションと構文に基づくパターンを用いて用例を提案する。例えば、ユーザが “We discussed” と書くと、システムは “discuss” という単語が使用されているパターンである “discuss in something” や “discuss with something” などを含む用例を表示する。一方、我々のシステムは、学習者の書いた誤った表現に対応する正しい例を提案することができる。

Sketch Engine [Kilgariff 04] は、単語に関連付く文法構成要素をソーラスの情報と共に表示する。前述のように、我々のシステムは、生コーパスから抽出された母語話者の書いた正しい例と共に、学習者コーパスを使用して学習者の書いた誤りを含む例を提示する。

3. 日本語学習者向けの文法誤り検出機能付き作文用例検索システム

文法誤り検出機能と用例検索機能を組み合わせた日本語学習者向けの用例検索システムを構築した。本システムは、日本語学習者であるユーザの書いた文が誤りを含む可能性があるとして仮定し、ユーザの入力に文法誤り検出を施す。誤りが検出された場合、その誤りをクエリとして誤用例文検索を行う。それ以外の場合はユーザの入力全文がクエリとして正用例文検索によって処理される。

3.1 データセット

本研究では Lang-8 コーパス [水本 13] を使用した。Lang-8 コーパスは学習者の作文であるエッセイとその添削、エッセイ ID、ユーザ ID、学習言語タグ、母語タグからなる。学習言語のタグが Japanese のエッセイ数は 185,991 編である。すべての学習者の文には 1 つ以上の添削文が付与されている。添削文が 1 つの場合は 1 つの学習者文と 1 つの添削文を文対とした。1 つの学習者文に 2 つ以上の添削文が付与されている場合は、それぞれの添削文と学習者文を文対とした。本研究では、Lang-8 コーパスに含まれる、日本語学習者が書いた添削前の文である誤用例と日本語母語話者が添削した文である正用例のペア 140 万文対を抽出した。Lang-8 の訂正データには、余分なコメントなどが含まれているものがあるため、抽出したデータの内誤用例の文字数が 100 文字以下であり、かつ、誤用例と正用例間の編集距離が 7 文字以下である 72 万文対を用いた。

*2 http://ngc2068.tufs.ac.jp/corpus_ja/

表 1: 正誤ラベルの例. ラベル c はターゲットの単語が正しいことを表し, ラベル i はターゲットの単語が誤っていることを表す.

正用文	いま	,	ぼく	は	がっ	こう	が	とても	いそがしい	です	よ	。
誤用文	いま	,	ぼく	は	がっ	く	が	とても	いそがし	です	よ	。
ラベル	c	c	c	c	c	i	c	c	i	c	c	c

3.2 日本語学習者の文法誤り検出

本研究では文法誤り検出を系列ラベリングタスクとして扱う. Rei ら [Rei 16] に基づいて入力文の各単語に正誤ラベルを割り当てた. 誤用文と正用文の文対で単語アラインメントをとり, 一致しているところを c , 一致していないところを i とした. 単語が削除されている場合はその直後の単語に i が付けられる. 表 1 はラベル付けを行った例である.

Rei ら [Rei 16] と同様に, Bi-LSTM を用いて文法誤り検出モデルを構築する. 我々のコードは GitHub にて公開している. 図 2 にモデルの構造を示す. システムは長さ T の単語系列 $[w_1 \dots w_t \dots w_T]$ を入力として受け取り, 最終的に各単語に対するラベルを予測する. まず, すべての単語 w_t は単語ベクトル e_t^w と文字ベクトル e_t^c に変換される. 文字ベクトルは, 各文字を文字分散表現に変換した後に Bi-LSTM に入力し, その先頭と文末の隠れ層を結合することで作成する. これらのベクトルは, 学習時に更新される. また, この文字ベクトルを作成する LSTM を文字 LSTM と呼ぶ. そして, e_t^w と e_t^c を結合することで t 番目の入力ベクトル $\tilde{x}_t$ を作成する. 入力ベクトルは Bi-LSTM を用いて以下のように隠れ層 h_t を計算する.

$$\vec{h}_t = \text{LSTM}(\tilde{x}_t, \vec{h}_{t-1}) \quad (1)$$

$$\overleftarrow{h}_t = \text{LSTM}(\tilde{x}_t, \overleftarrow{h}_{t+1}) \quad (2)$$

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (3)$$

$\vec{h}_t$ は順方向 LSTM, $\overleftarrow{h}_t$ は逆方向 LSTM, h_t は両方向の隠れ層を結合したものである. また, $\vec{h}_0 = \overleftarrow{h}_0 = \mathbf{0}$ である. 次に, 全結合層を使用して Bi-LSTM と出力層の次元数の差を調整するための追加の隠れ層 d_t を計算する.

$$d_t = \tanh(W_d h_t) \quad (4)$$

ここで W_d は出力重み行列である. そして, 出力層と softmax 関数を用いて予測を行う.

$$P(y_t | w_1 \dots w_T) = \text{softmax}(W_o d_t) \quad (5)$$

ここで W_o は出力重み行列であり, y_t は予測ラベルである.

3.3 用例検索アルゴリズム

用例のデータベースには 3.1 節で構築したデータを用いる. ユーザの検索クエリを受け取り, データベースからその文に関連する誤用例と正用例の文対を抽出

し, 表示する. 検索クエリは文または複数単語を想定している.

- (1) 3.2 節で説明した文法誤り検出システムを用いて検索クエリの誤りを検出する.
- (2) 誤りの検出結果に応じて, 用例を提示する.
 - (a) 誤りが検出された場合, その単語を誤用例に含む文を抽出する. 誤りが複数箇所ある場合は全ての誤りに対して個別に抽出し, 3.4 節で説明する方法で誤用例をリランキングし, 正用例と共に表示する.
 - (b) 誤りが検出されなかった場合, 事前に単語単位に分かち書きされた正用例と, 単語単位での比較を行い, 両者に一致する箇所が存在するものを抽出する. このとき, 一致単語数が多い用例ほど上位に表示する. 更に, 一致する単語パターンが同じ正用例が複数ある場合は, 3.4 節で説明する方法でリランキングを行う.

3.4 例文検索におけるリランキング

文法誤り検出結果による検索クエリによって抽出された例文のリランキング方法は, ユーザが入力した検索クエリの文ベクトルと例文の文ベクトルのコサイン類似度の降順である. 文ベクトルの生成方法はいくつか考えられるが, 本研究では以下の代表的な 2 種類の手法を試みた. 単語の意味を分散表現化することが可能な word2vec [Mikolov 13] を用いた手法と, 近年自然言語処理で注目されている既存の様々なタスクに転移学習可能な大規模事前学習モデルである Bidirectional Encoder Representations from Transformers (BERT) [Devlin 19] を用いた手法を使用した.

i. word2vec

3.1 節と同様の前処理を行った Lang-8 コーパスの添削文側のデータを用いて学習した word2vec モデルを使用して, 各単語ベクトルの分散表現を算出し, その平均を文ベクトルとした.

ii. BERT

日本語 Wikipedia で事前学習済みのモデル [柴田 19] をファインチューニングを行わずそのまま使用する. このモデルの学習時の入力テキストは Juman++ (v2.0.0-rc2) [Morita 15] で形態素解析を行い, さらに Byte Pair Encoding (BPE) [Sennrich 16] を適用しサブワードに分割している. この設定に合わせて本研究においても入

図 2: 文法誤り検出モデルの構造。「私の元気です」という文が入力された場合の例。*3

カテキストは Juman++ による分かち書きと BPE によるサブワード化を行い、トークンの系列に変換する。この系列を事前学習済みのモデルに入力し、最終層の全単語の隠れ層の平均を文ベクトルとした。

4. 評価実験

4.1 日本語学習者の文法誤り検出

文法誤り検出の事前実験には 3.1 節で説明したデータを用いた。このデータを学習データ 720,000 文と開発データ 1,000 文、評価データ 1,000 文の 3 つに分割して利用した。文法誤り検出の正解データは、誤用文を正用文と比較し、一致箇所にラベル c を、不一致箇所にラベル i を振る動的計画法によって [Kaneko 17] 自動で生成したものを用いた。評価は単語単位で行う。正確なフィードバックのほうがカバレッジの高い誤り検出より学習効果のあることが知られているため [Nagata 10], precision を重視する $F_{0.5}$ で評価した。

i. 実験設定

形態素解析器には MeCab (ver. 0.996)*4 を使用し、辞書は UniDic (ver. 2.2.0) を使用した。文ベクトルを作るために gensim*5 を使用した。ハイパーパラメータの設定は、単語分散表現と単語 LSTM の次元数は 300、文字分散表現と文字 LSTM の次元数は 100 とした。adadelta (学習率: 1.0) でバッチサイズを 64 文として最適化した。

*3 簡略化のため 3 番目の単語である「元気」にのみ単語ベクトル化と Bi-LSTM による文字ベクトル化、そしてその結合の処理を描いているが、実際には全ての単語でこの処理が行われる。

*4 <https://github.com/taku910/mecab>

*5 <https://github.com/RaRe-Technologies/gensim>

表 2: 日本語文法誤り検出モデルの精度の比較。

モデル	precision	recall	$F_{0.5}$
SMT モデル	0.599	0.121	0.333
ニューラルモデル	0.615	0.304	0.511

比較システムとして水本ら [水本 13] の統計的機械翻訳 (SMT) システムを再実装した。単語-単語対応モデルを使用した。SMT システムは誤り訂正モデルのため出力が訂正文であるので、ニューラルモデルと同様に評価するためにシステムの出力文をラベルに変換したものを最終出力とした。ラベルへの変換方法は正解データの生成と同様に、出力文を正用文と比較し、一致箇所にラベル c を、不一致箇所にラベル i を振る動的計画法によって行った。Minimum Error Rate Training (MERT) [Och 03] を用いて $F_{0.5}$ で最適化した。

ii. 実験結果

各モデルにおける日本語文法誤り検出精度を表 2 に示す。precision と recall, $F_{0.5}$ の全てにおいてニューラルモデルのほうが従来手法より上回っており、ニューラルモデルのほうが優れていることが分かる。

表 3 に誤りタイプごとの true positives と false negatives, false positives の数を示す*6。人手で評価データ 1,000 文に対するシステム出力を true positives, false negatives, false positives, true negatives に分類し、さらに true positives と false negatives の出力にはそれぞれ

*6 false positives は誤りでない単語を誤りと誤認識した箇所のため、誤りタイプは存在しない。

表 3: ニューラルモデルにおける true positives (TP) と false negatives (FN), false positives (FP) の数.

誤りタイプ	TP	FN	FP
全て	294	263	106
助詞選択	75	60	—
混合	16	38	—
語彙選択	22	77	—
脱落	33	18	—
誤形成	53	14	—
余剰	40	27	—
発音	55	25	—
その他	0	4	—

表 4: true positives (TP) と false negatives (FN), false positives (FP) の例. 下線は正解の誤り箇所, 山括弧はシステムが誤りと判定した箇所を示す.

TP	おねがい、しあわせに <u>〈なる〉</u> !
FN	定刻になると、徳川 <u>家安</u> が出てきます。
FP	これ、私の名刺 <u>〈で〉</u> ございます。

誤り検出した箇所と誤り検出できなかった箇所の誤りタイプを手で付与した. 誤りタイプは学習者文に基づいた誤りタイプであるため, 学習者文が正しい場合は誤りタイプは付けられない. 助詞選択, 発音, 誤形成の誤りは検出率が高く, 語彙選択と混合の誤りは検出率が低かった. 助詞選択は全ての誤りの大部分を占めるため, false negatives の数も多くなっている.

表 4 にニューラルモデルにおける true positives と false negatives, false positives の例を示す. ニューラルモデルはニューラルネットワークを用いているため, 表 4 の true positives の例のような長距離関係を考慮した誤りを検出することができる. この例は文頭に「おねがい」とあるため, 未来形ではなく願望を表す文であり, 「なる」ではなく「なりたい」が適切な表現となる. 一方で, 表 4 の false negatives の例のような誤りはデータ不足のため, 検出することができない. また表 4 の false positives のような例もデータ不足により誤って検出してしまう. 「名刺」を使った用例はコーパス中に 2 件しかなく, どちらも「で」ではなく「に」を使ったものであった. そのためシステムは誤って検出したものと考えられる.

4.2 検索システムの評価実験

誤用例表示と誤り検出の有用性を確かめるため, 比較システムを用いて, 内的評価と外的評価の 2 つの方法で評価実験を行った.

§1 内的評価

i. 評価設定

ここでは提案する検索システムの有用性を定量的に調べるために, ランダムな検索クエリを与えたときの $\text{precision}@10^{*7}$ を測り, 比較システムと比較した.

比較システムは 3.1 節で構築したデータを用い, 誤り検出と並び替えは行わず, 誤用例側で検索をかけるか正用例側で検索をかけるかを学習者が任意で選択する必要のあるものを使用した. それ以外はすべて提案システムと同様の設定を適用する. 学習者の誤りを自動で検出し, 誤用例側で検索をかけるか正用例側で検索をかけるかをシステムが自動で判断することの有用性を確かめるため, 正用検索を選択した場合と誤用検索を選択した場合の両方を比較実験に使用した.

ランダムな検索クエリとして, Lang-8 コーパスに含まれる学習者文のうち, 3.1 節の前処理で除外されたものに含まれるフレーズを使用する. 実際にユーザが使用する際は, 誤用と正用の双方とも検索することが考えられるため, 検索クエリには誤用フレーズと正用フレーズの双方を用いる. どちらも 55 フレーズずつ, 計 110 フレーズを抽出し *8 , 使用した. ここで, Lang-8 コーパスは誤用文とその訂正文のペアを持つため, 誤用フレーズにはそれが訂正された訂正フレーズが存在することに留意されたい.

ii. 評価方法

クエリを比較システム (正用), 比較システム (誤用), 提案システムで検索し, その正解のフレーズに完全一致する用例が得られたフレーズの数を数え, $\text{precision}@10$ を測った. 学習者が誤用を入力した場合, その学習者にとって必要なのは入力した誤用が訂正された用例であるため, 検索結果の適・不適は訂正文におけるフレーズが含まれているか否かで決める. ここで, 誤用フレーズの場合, それと対になる訂正されたフレーズを正解フレーズとし, 正用フレーズの場合はフレーズそのものを正解フレーズとし, $\text{precision}@10$ を測った. フレーズを検索した際, 検索結果に正解フレーズが含まれている用例があったら正解であり, そうでない場合不正解と判断した.

iii. 評価結果

誤用フレーズの誤りタイプを混同, 語彙選択, 脱落, 誤形成, 余剰, 発音, その他の 7 種類に分類した. 表 5 は誤りタイプ別のフレーズ例である. 各システムの誤りタイプ別の頻度と precision を表 6 に示す.

1) 比較システム (正用)

正しいフレーズでは最も高い precision となったが, 誤りフレーズの検索クエリは誤り文字列であるため, 正用例側で検索をかけても一致するものがなく, 正解の

*7 ランキングの上位 10 件における適合率.

*8 誤りフレーズはランダム抽出された文から誤用部分を人手で抜き出し, 正用フレーズも同様にランダム抽出された文から正用部分を人手で抜き出した.

表 5: 誤りタイプ別テストフレーズ例.

誤りタイプ	誤り文字列	正解文字列	比較システム (正用)	比較システム (誤用)	提案システム
脱落	おねさん	おね <u>え</u> さん	×	×	✓
脱落	ニュージーランド	ニュー <u>ジ</u> ーランド	×	✓	✓
余剰	みんなさん	みなさん	×	✓	✓
余剰	大体に	大体	×	×	✓
誤形成	疑問 <u>を</u> して	疑問 <u>に</u> 思 <u>っ</u> て	×	×	✓
誤形成	驚 <u>い</u>	驚 <u>き</u>	×	✓	×
混同	が <u>も</u> らえる	し <u>か</u> も <u>ら</u> えない	×	×	×
混同	稼 <u>ぐ</u>	稼 <u>い</u> だ	×	✓	×
語彙選択	ち <u>さ</u> い	少 <u>な</u> い	×	✓	×
語彙選択	助 <u>け</u> られる	で <u>き</u> る	×	×	✓
発音	し <u>ま</u> し <u>だ</u>	い <u>ま</u> し <u>た</u>	×	×	×
発音	死 <u>ん</u> ち <u>ゃ</u> う	死 <u>ん</u> じ <u>ゃ</u> う	×	✓	✓
その他	ハ <u>ウ</u> ス	家	×	✓	✓

表 6: 内的評価実験結果. 「比較・正」は比較システムの正用検索, 「比較・誤」は比較システムの誤用検索, 「提案」は提案システムを表す.

誤りタイプ	頻度	precision@10		
		比較・正	比較・誤	提案
誤り全て	55	0.00	0.45	0.44
混同	19	0.00	0.37	0.32
語彙選択	16	0.00	0.38	0.19
脱落	8	0.00	0.75	0.75
誤形成	6	0.00	0.40	0.67
余剰	3	0.00	0.67	1.00
発音	2	0.00	0.50	0.50
その他	1	0.00	1.00	1.00
正例	55	0.90	0.15	0.85
全ての平均	110	0.45	0.30	0.65

用例を得ることができなかったため, precision は 0.00 となった.

2) 比較システム (誤用)

誤用例側で検索をかけるため, 正しいフレーズで検索した場合には正解の用例はほとんど出てこなかった. 誤りフレーズ検索では最も高い precision が得られた. 検索クエリに一致した誤り文字列を含む誤用例と, 対になる正用例を検索できる.

3) 提案システム

提案システムは, 誤り全体の precision と正用の precision はどちらも比較システムよりもわずかに低いが, 比較システムはユーザが手動で誤用検索と正用検索を切り替えなければならないのに対し, 誤り検出機能により入力したクエリが正しいか誤りであるかを判断して, 自動で適切なほうを検索するため, 検索を切り替える負担が存在しない. そして, 誤りフレーズ検索と正し

いフレーズの検索の両方を含めた全体の平均 precision は提案手法であるこのシステムが最も高く, 検索システムとして有用であると考えられる.

比較システムと比べて, 提案システムは誤形成の precision が高かった. この誤りタイプは学習データ中に類似した形のものが多く含まれるため, 誤り検出の精度が良く, さらに, 類似度でソートしているため, 入力クエリと意味的に近い文が上位に来るため, その誤りを正しく直された文を表示できるからだと考えられる. また, 脱落や余剰などの分かりやすい誤りは, 問題なく誤り検出が機能し, 比較的高い precision で同定することができた.

一方で, 語彙選択が検索しづらかった. 提案システムは検索結果をコサイン類似度で降順にソートしているため, 入力クエリと似たような語を使った文が上位にくる. そのため入力クエリと全く異なる語彙が正解だった場合, そのコサイン類似度は低くなり, 提案システムは適切な答えを出すことができない. また, 文法誤り検出の精度が十分でないため, 誤っている入力クエリが正しいと判断されることも多く, その場合自動的に正用検索になるため, 正しく直された文が表示されない. 今後誤り検出の精度が上がることで用例検索システムの精度も上がることが期待される.

§2 外的評価

i. 評価設定

被験者実験では, 提案システムを以下の3つの設定で使用する. システム3が提案手法である. 例文のリランキングに word2vec を使用したシステムと BERT を使用したシステムの両方で内的評価を行った結果, 全く同じ結果となったため, 検索スピードの速い word2vec を使用したシステムを外的評価で使用する. 全てのシステムで 3.1 節で構築したデータを用い, 以下に示す設定以外は全て提案システムと同様の設定を適用する.

表 7: 外的評価のための日本語作文の質問.

No.	質問
1	あなたの住んでいる町を紹介.
2	夏と冬とどちらが好きですか. 理由も.
3	日本語の難しいところ.
4	テレビニュースと新聞は何が違うのか.
5	海外で経験してみたいこと.
6	自由な時間があったら, 何をしてみたいか.
7	自分の国の魅力を紹介.
8	嘘をつくのはいいことか.
9	あなたが健康のためにしていること.
10	大学生活で一番楽しかったこと.

表 8: 外的評価結果. 表の 1, 2, 3 はそれぞれシステム 1, システム 2, システム 3 を示す.

被験者	JLPT	1	2	3
A	N1	14	20	21
B	N1	26	27	29
C	N1	15	16	16
D	N1	28	25	26
E	N2	22	25	25
F	N1	20	23	28
平均		20.8	22.7	24.2

- システム 1: 誤り検出なし, 正用例のみ表示
- システム 2: 誤り検出なし, 正用例と誤用例を表示
- システム 3: 誤り検出あり, 正用例と誤用例を表示

まず, システム 1 を使った作文とシステム 2 を使った作文の点数を比較して, 誤用例と正用例の文対表示の有用性を確認する. 次に, 文法誤り検出機能により, 提案システムが学習者の書いた誤りを含む文をどれだけ校正させられるかを確かめるため, システム 2 を使った作文とその作文をシステム 3 を使って直した後の点数を比較する.

ii. 評価方法

日本語を母語としない日本語学習者 6 人に日本語作文の問題を解答してもらい, その点数で評価を行う. 被験者は全員, 情報工学を専攻する母語が中国語の大学院生である. 日本語のレベルは 5 人が日本語能力試験 (JLPT) N1, 1 人が N2 である *9.

作文の問題は全部で 10 問あり, 5 問ずつ A テーマと B テーマに分かれている. 被験者にはまず, A テーマの問題と B テーマの問題をそれぞれシステム 1 とシステム 2 を使用して書いてもらう. テーマによる評価の偏りを防ぐため, 被験者のうち半数は A テーマと B テーマを入れ替えて行う. 次に, システム 2 で書いた作文

表 9: 外的評価での誤りタイプ別の頻度と正解率.

誤りタイプ	頻度	#	正解率
誤り全て	44	9	0.20
混同	6	1	0.17
語彙選択	14	4	0.29
脱落	7	1	0.14
誤形成	6	2	0.33
余剰	9	1	0.11
発音	2	0	0.00
その他	0	0	0.00

表 10: 外的評価の作文の例. 上段がシステム 3 でうまく直せた例, 下段が直せなかった例を示す. 太字は変更された箇所を示す.

システム 2	夏に花火大会や祭りが見えます.
システム 3	夏に花火大会や祭りが見られます.
システム 2	大都市から離してよかったです.
システム 3	大都市から離してよかったです.

をシステム 3 を使用して校正してもらう. また, 評価基準を統一するため, 1 問につき 3 文で書くよう指示を付けた. 表 7 に実験に使用した日本語作文の質問を示す.

採点は被験者に書いてもらった文章を基に減点法で行った. 持ち点 1 人 30 点で, 1 つの文法誤りにつき 1 点引いていき, 各システムごとに合計し, 最終的な点数を計算した.

iii. 評価結果

結果を表 8 に示す. 点数の平均はシステム 1 が 20.8, システム 2 が 22.7, システム 3 が **24.2** と提案手法が最も高く, システム 2 と同じ点数も含め, 6 人中 5 人が提案手法で最も高い点数を出した.

表 9 はシステム 2 を使って書いた作文をシステム 3 を用いて校正した時に誤りを訂正できた割合を表す *10. 内的評価と同じように, 誤形成が最も高い割合で訂正できていた. 内的評価とは異なって語彙選択もよく訂正できていたが, 語彙選択の正解率の内訳は内容語が 0.17, 機能語が 0.38 となっているので, 機能語が比較的高い割合で訂正できている. 内的評価では助詞などの機能語は評価の対象ではないため分からなかったが助詞などの機能語の選択誤りには対応できることが分かる. 表 10 は実際の作文の例である.

5. おわりに

本研究では, 誤用例文検索の有用性に着目し, 正用例付き誤用例文の表示を行った. さらに, 本システムでは誤り検出を自動で行うことによって言語学習者が

*9 日本語能力試験にはレベルが N1 から N5 まであり, N1 が最も高いレベルで幅広い場面で使われる日本語を理解することができるレベル, N2 が日常的な場面 (N3) で使われる日本語の理解に加え, 日常的な場面 (N3) より幅広い場面で使われる日本語をある程度理解することができるレベルとされている.

*10 このとき, 誤り数と誤りの訂正数は人手で数えた.

気づいていない誤りを認識し、誤用例を提示する。この機能により、学習者は誤りを把握して解釈し、訂正するか否かを選択できる。

日本語学習者向けの文法誤り検出機能付き誤用例文検索システム実験により、内的評価と外的評価で共に提案手法が最も優れていることを示した。

謝 辞

Lang-8 のデータ使用に際して、快諾してくださった株式会社 Lang-8 社長喜洋洋氏に感謝申し上げます。この研究の一部は JSPS Grant-in-Aid for Scientific Research (C) Grant Number JP19K12099 によって支援されています。

◇ 参 考 文 献 ◇

- [Chen 12] Chen, M.-H., Huang, S.-T., Hsieh, H.-T., Kao, T.-H., and Chang, J. S.: FLOW: A first-language-oriented writing assistant system, in *Proceedings of the ACL 2012 System Demonstrations*, pp. 157–162 (2012)
- [Devlin 19] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proceedings of NAACL*, pp. 4171–4186 (2019)
- [Kaneko 17] Kaneko, M., Sakaizawa, Y., and Komachi, M.: Grammatical error detection using error- and grammaticality-specific word embeddings, in *Proceedings of IJCNLP*, pp. 40–48 (2017)
- [Kilgariff 04] Kilgariff, A., Rychly, P., Smrž, P., and Tugwell, D.: The Sketch Engine, in *Proceedings of EURALEX*, pp. 105–116 (2004)
- [李 08] 李 在鎬, 浅尾 仁彦, 濱野 寛子, 佐野 香織, 井佐原 均: タグ付き日本語学習者コーパスの開発, 言語処理学会第 14 回年次大会, pp. 658–661 (2008)
- [Mikolov 13] Mikolov, T., Chen, K., Corrado, G., and Dean, J.: Efficient estimation of word representations in vector space, in *Proceedings of ICLR* (2013)
- [水本 13] 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)
- [Morita 15] Morita, H., Kawahara, D., and Kurohashi, S.: Morphological analysis for unsegmented languages using recurrent neural network language model, in *Proceedings of EMNLP*, pp. 2292–2297 (2015)
- [Nagata 10] Nagata, R. and Nakatani, K.: Evaluating performance of grammatical error detection to maximize learning effect, in *Proceedings of COLING*, pp. 894–900 (2010)
- [Nishina 14] Nishina, K., Hodošček, B., Yagi, Y., and Abekawa, T.: Construction of a learner corpus for Japanese language learners: Natane and Nutmeg, *ALA*, Vol. 4, No. 2, pp. 37–51 (2014)
- [Och 03] Och, F. J.: Minimum error rate training in statistical machine translation, in *Proceedings of ACL*, pp. 160–167 (2003)
- [Rei 16] Rei, M. and Yannakoudakis, H.: Compositional sequence labeling models for error detection in learner writing, in *Proceedings of ACL*, pp. 1181–1191 (2016)
- [Sennrich 16] Sennrich, R., Haddow, B., and Birch, A.: Neural machine translation of rare words with subword units, in *Proceedings of ACL*, pp. 1715–1725 (2016)
- [柴田 19] 柴田 知秀, 河原 大輔, 黒橋 慎夫: BERT による日本語構文解析の精度向上, 言語処理学会 第 25 回年次大会, pp. 205–208 (2019)
- [Yen 15] Yen, T.-H., Wu, J.-C., Chang, J., Boisson, J., and Chang, J.: WriteAhead: Mining grammar patterns in corpora for assisted writing, in *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pp. 139–144 (2015)

[担当委員: 狩野 芳伸]

2020 年 2 月 6 日 受理

著 者 紹 介

新井 美桜

2017 年帝京大学外国語学部外国語学科卒業。2018 年より首都大学東京システムデザイン研究科博士前期課程に進学。2020 年博士前期課程修了。

金子 正弘

2016 年北見工業大学工学部情報システム工学科卒業。同年、首都大学東京システムデザイン研究科博士前期課程に進学。2018 年博士前期課程修了。同年、同研究科博士後期課程に進学。2019 年より日本学術振興会特別研究員 (DC2)。言語処理学会第 23 回年次大会若手奨励賞受賞。

小町 守 (正会員)

2005 年東京大学教養学部基礎科学科科学史・科学哲学分科卒業。2008 年より日本学術振興会特別研究員 (DC2) を経て、2010 年奈良先端科学技術大学院大学情報科学研究科修了。博士 (工学)。同年より奈良先端科学技術大学院大学情報科学研究科助教を経て、2013 年より首都大学東京システムデザイン学部准教授。大規模なコーパスを用いた意味解析および統計的自然言語処理に関心がある。言語処理学会 20 周年記念論文賞、言語処理学会第 14 回年次大会最優秀発表賞、情報処理学会平成 22 年度山下記念研究賞、2010 年度本学会論文賞等を受賞。情報処理学会、言語処理学会、ACL 各会員。