変更履歴に着目した論文校正支援方法の提案

2017SE046 松井亮介　2017SE099 渡辺開斗
指導教員：蜂巣吉成

1　はじめに

大学の論文作成作業では，学生が論文を執筆して教員へ提出し，教員は添削を行い学生へ論文を返却する．論文を完成させるまでこの作業を何度も繰り返す．

添削作業の問題点として，教員が添削する論文の数が増えると，添削作業量が多くなることがある．文章表現の指摘といった複数の論文に対して同じような修正指示をする作業が多くなると，教員は論文の本質的な内容に対する指摘に時間をかけることができない．

既存の添削支援方法として，ルールベースの文章校正方法がある．ルールベースの場合，事前にルールを定めなければならない上に，ルールの更新を適宜行う必要がある．

本研究では，変更履歴に着目した論文の校正支援方法を提案する．論文を入力すると，表現として誤りが含まれていると思われる文を発見して利用者に提示することを目指す．そのために，過去の論文の変更履歴からデータを収集し，蓄積されたデータから未知の論文に対して評価をする．

変更履歴に着目した理由として，変更前が誤りを含む表現，変更後を誤りが修正された文と考え，修正箇所を蓄積することで，人がルールを記述せずともルールを構築できるという点がある．構築されたルールは変更履歴を増やすほど充実するので，校正の精度もデータの数に応じて向上することが見込める．

本研究では，論文の添削前後の文の対応関係を特定し，変化した内容を変更履歴として用いて，修正が必要な文の判定を行う方法を提案する．そのために，次の研究課題を挙げ，これに沿ったアプローチを提案する．

1. 添削前後の文の対応関係と，変化の種類を特定する
2. 単語レベルで校正を支援する
3. 単語レベルで検出できない誤り文を機械学習で特定する

添削前を誤り，添削後を正しいものとみなし，正しい文と誤り文を確保する．その後，単語レベルの表現の誤り検出を形態素の比較によって行う．それで検出できないような，文節や文法表現についての誤り検出を，CNN や RNN 分類器を用いた機械学習によって実現する．

本研究の妥当性と課題点を確認するために，実際の論文を用いてプロセスごとに実験を行い，評価・考察をした．

2　既存ツール・関連研究

textlint[1] はルールベースの文章校正方法で，手動でルールを作成し，機械的に文を評価する．教員や学問分野ごとに適切とされる文章表現は異なるが，それぞれに合わせたルールを設定することは膨大な手間となる．本研究では，変更履歴に着目してルール作成と更新作業を自動化することで，こういった問題点の解消を図った．

山田らの研究 [2] は文書作成履歴に着目し，過去に作成した文をすべて正しい校正知識として蓄積する．本研究での修正対象が文章表現であるのに対し，修正対象は文章の表層の誤りである．入力された文章と過去の類似した文章を比較し，差分を取ることで表層の誤り検出を実現する．ルールとなる校正知識は過去の論文との違いに依存するので，ルールを定義することは想定されない．

坂本らの研究 [3] は，文章表現規則を定義し，作成された文に適用して不都合箇所を検出する．ルールベースで指摘できる文について多くの指摘をすることができる．文章表現規則は技術文章ルールと文章一般ルールの観点から事前に定義される．したがってルールの変更や新規ルールの作成は考慮されない．

3　論文校正支援方法の提案

本研究の最終的な目標は，文章表現の修正が必要な文の検出，および検出ルール構築の自動化である．ここでの文章表現として，単語，文法を想定する．添削前後の文の変化を特定し，添削前後で変化した単語や品詞関係をデータとして蓄積することで検出ルールの構築を行う方法を考えた．そこで，論文の本文が添削前後でどう変化したかを 1 文ごとに知る必要がある．最初に添削前後で対応している文と変化の種類を特定する．そのあと，対応の取れた文から単語の変化，および品詞間の変化を調べる．これらを考慮して次の 3 つのプロセスを考案した．

1. 文単位での変化の種類の特定
2. 形態素の比較による単語レベルでの評価
3. 機械学習による文レベルでの評価

文単位における変化の種類の特定では，文の類似度を定義し添削前後の文でマッチングを行う．このプロセスで，ルール構築に必要となる添削前後の文の対応関係と変化の種類を明らかにする．形態素の比較による単語レベルでの評価では，変更履歴から変化している単語の辞書を作成することで，単語表現の誤りを検出する．機械学習による文レベルでの評価では，文法表現や品詞間の関係性に着目して，機械学習を用いることで文を調べる．添削前を誤り，添削後を正しい文として学習させ，文を正しい/誤りに分類する問題として文検出を行う．

3.1　文単位での変化の種類の特定

本研究では，文単位での変化の種類を次のように定義する．

置換　文中の一部分または複数の部分的な変化

分裂　添削前に存在する 1 つの文が，添削後で複数の文に分かれている変化

統合　添削前に存在する 2 つの文が，添削後で 1 つの文にまとめられている変化

追加　添削前の文章内に存在しない文が添削後に新たに加わっている変化

削除　添削前に存在している文が添削後に消去されている変化

変化の種類を特定するには，添削前後の文の対応を取る必要がある．そのために，添削前後の文を総当たりして，全てのペアに対して類似度を計算する．本研究では，類似度を計算するにあたって，添削前後で共通する部分を求めて数値化することができる，LCS[*1]アルゴリズムを用いる．算出した類似度に閾値などの条件を設け，添削後の文が添削前に対して置換されているものを検出する．

3.2　形態素の比較による単語レベルでの評価

3.1 節で置換と判定された文のペアから，良い単語/単語群と悪い単語/単語群を特定する．これをデータとして蓄積することで，未知の文に対して単語レベルの校正を可能にする．本研究では，添削されがちな単語レベルの間違いデータと修正案を辞書に保存する．検出対象としたい変化は，次の 2 つである．

(1) 1 つの独立した形態素が置換されている

(2) 連続した形態素がまとめて置換されている

単語レベルで変化を追うには，添削前後の単語同士の対応を取る必要がある．そこで置換判定されたペアの差分を形態素レベルで取る．差分を取ることで，単語同士の対応を取りつつ，変化している部分を抽出できる．差分を取った結果から，置換されている形態素/形態素群を，添削前のものは誤り，添削後のものは正解として辞書に保存する．

3.3　機械学習による文レベルでの評価

単語レベルの変化の評価では単語の表層の形に着目したが，その中でも接続詞や助詞などは文脈によって正しいかが決まるので，表層情報のみを用いる単語レベルでの指摘は難しい．そこで品詞情報や活用形情報を用いることで文節や文法表現についての評価を実現する．

機械学習を利用して，文の添削前後での変化内容を教師あり学習することで，文を正解と誤りの 2 クラスに分類し修正必要性を判定する．教師データは文単位での変化の種類の特定において判定された対応関係をもとに作成する．その際，添削前の文を誤り，添削後の文を正しいものとみなしてタグ付けする．その後，文をハッシュ化してベクトル表現にする．表層の形を参照するだけでは正しいか判断しにくい表現に対して，傾向や特徴が現れると考えたので，ハッシュ化には表層，品詞，活用形の情報を用いる．

作成したベクトルを機械学習アルゴリズムを用いた分類器に入力する．機械学習アルゴリズムには，自然言語処理の適性を考慮して，CNN[*2]と RNN[*3]の 2 つを採用する．

CNN では，入力したベクトルを embedding 層で各 ID の分散表現にエンコードする．それぞれの分散表現について，表層，品詞，活用形の 3 つをフィルターによって畳み込む．畳み込んで得られたフィルタにマックスプーリング処理を行う．次に 2 つずつ要素の畳み込みとプーリングを行い，出力層を全結合する．その後，さらに畳み込み処理を行うことで複数の形態素情報を畳み込めると考え 2 回の畳み込みを行った．

RNN では，入力したベクトルを embedding 層で各 ID の分散表現にエンコードする．それぞれの分散表現を RNN 層に入力する．RNN 層からの出力を全結合で出力層に連結する．

4　実験

4.1　実験 1：文単位での変化の種類の特定

4.1.1　目的

実験 1 は，文単位での変化の種類の判定での判定精度についての検証である．この実験で，提案した方法の置換判定精度を確認する．

4.1.2　方法

2020 年度の我々の研究室における卒業研究の中間発表予稿 A，B，C，D を用意し，それぞれにプロセス 1 を適用する．その結果と，手作業で作成した正解データを照合して，再現率・適合率・F 値を算出する．

4.1.3　結果・考察

表 1 は，実験 1 の結果である．

表 1　実験 1 の置換判定結果

論文	置換判定数	正答数	再現率	適合率	F 値
論文 A	183	135	0.81	0.74	0.77
論文 B	46	35	0.36	0.76	0.49
論文 C	55	49	0.45	0.89	0.60
論文 D	72	70	0.45	0.97	0.61

本稿で提案した方法で添削前後での文の対応関係を特定することができたが，分裂と統合の特定については，サンプル内に十分な例がなかったので検証ができなかった．今後サンプルをさらに増やして検証をする必要がある．

4.2　実験 2：形態素の比較による単語レベルでの評価

4.2.1　目的

実験 2 は形態素の比較による単語レベルでの評価についての検証である．この実験では，本稿で提案した方法を用いて辞書を作成し，実際にどのような単語が保存されたか，その単語の出現回数などを確認する．

[*1] Longest Common Subsequence：最長共通部分列

[*2] Convolutional Neural Network：畳み込みニューラルネットワーク

[*3] Recurrent Neural Network：再帰型ニューラルネットワーク

4.2.2 方法

提案した方法を実際に実行して，得られる辞書の単語と出現回数の確認を行う．この実験では，我々の中間発表で作成した予稿の文を用いて実験する．

4.2.3 結果・考察

表 4.2.3 は，実験 2 の結果の一部である．

表 2 実験 2 の結果

添削前	添削後	出現回数
文章	文	32
判断	判定	3
チェックする	調べる	3
本項	本稿	3
することで	して	2
チェック	評価	2
に着目	を対象と	1
したい	する	1
ため	ので	1

本稿の方法によって表層の単語の変化を辞書に保存することができた．1 回のみ出現の変化が多く保存されたが，それらはすべて一般的な変化とは呼べない変化であったことから，出現回数でフィルタリングを行うなど処理を加える必要がある．「文章」が「文」に変化したものについては，表現修正として起こったものではあるが，他の論文でも一般的に出現するとは言えない．このような変化は，論文ごとに作成した辞書をすべて統合したときに，出現回数は相対的に少なくなる．そこで何回の出現があったらルールとして適用するかを検証する必要がある．

4.3 実験 3：機械学習による文レベルでの評価

4.3.1 目的

実験 3-1，3-2 は機械学習による文の評価での判定についての検証である．実験 3-1 は予備実験として，我々が想定できる例でのテストケースを作成し，誤りを学習できるかを検証する．実験 3-2 では，実際の論文に近い状態で誤りを学習できるかを検証する．学習できたかどうかの基準として，正答率 70% を目安とする．CNN 分類器と RNN 分類器の 2 つの分類器を用いて精度を比較し，どちらが適切な学習アルゴリズムかを判断する．

4.3.2 方法

実験 3-1 では特定の表現を含んだ文を集めた教師データを作成して分類を行い，その判定精度を観察する．実験 3-2 では，実験 1 でも使用した過去の論文すべてを用いて分類を行い，その精度を観察する．それぞれの実験で CNN と RNN を用いて検証する．用意したデータのうち無作為に抽出した 70% を学習に使い，残りの 30% を分類検証に用いる．表層，品詞，活用形でハッシュ化したものをハッシュ化 1，活用形情報を除いたものをハッシュ化 2 とする．ここで用意したテストパターンと対象の表現は表 3 の通りである．表 3 の 1 列目は次のパターンに対応している．

パターン 1 1 文の中に 1 種類のみの変更がある文を集めた教師データでの分類実験

パターン 2 1 文の中に 2 種類以上の変更がある文での分類実験

パターン 3 パターン 1，2 を混ぜた文での分類実験

表 3 テストパターン一覧

パターン	番号	対象とする表現
1	1	「することができる」(冗長な表現,「できる」がよい)
	2	「ため」(理由，目的が曖昧な表現)
	3	体言止めで終わる文
	4	「〜し」が連続する文
	5	「〜たり」が 1 つだけの文
	6	「適用」,「適応」の誤用
	7	主語述語非対応
	8	1-1 から 1-7 まですべての文
	9	1-8 のうち誤りが 1 箇所の文
2	1	パターン 1 の表現が 1 文に 2 箇所以上現れる文
3	1	1-9 と 2-1 のすべての文
	2	パターン 1,2 すべての文

4.3.3 結果・考察

実験 3-1 の，ハッシュ化 1，2 で CNN 分類器を用いた分類結果はそれぞれ表 4 および表 5，ハッシュ化 1 で RNN 分類器を用いた分類結果は表 6 に示す．

実験 3-2 の，ハッシュ化 1，2 で CNN 分類器を用いた分類結果はそれぞれ表 7 および表 8，ハッシュ化 1 で RNN 分類器を用いた分類結果は表 9 に示す．

表 4 ハッシュ化 1 による CNN での実験 3-1 の結果

番号	学習数	検証数	正答率	再現率	適合率	F 値
1-1	51	23	0.74	0.92	0.69	0.79
1-2	74	33	0.85	0.94	0.79	0.86
1-3	109	47	0.96	1.00	0.93	0.96
1-4	225	97	0.72	0.37	0.5	0.42
1-5	42	19	0.21	0.14	0.1	0.12
1-6	124	54	0.19	0.19	0.19	0.19
1-7	54	24	0.38	0.29	0.44	0.35
1-8	686	294	0.51	0.32	0.37	0.34
1-9	684	294	0.52	0.46	0.47	0.46
2-1	252	109	0.59	0.46	0.59	0.52
3-1	937	402	0.65	0.55	0.64	0.59
3-2	1623	696	0.79	0.73	0.80	0.76

実験 3-1 において，CNN を用いた場合，ハッシュ化 1 ではパターン 1-1 から 1-4 と 3-2，ハッシュ化 2 ではさらにパターン 2-1 と 3-1，RNN の場合はパターン 1-1，1-3，1-7 で正答率が 70% を超えている (1-7 はハッシュ化 2 の正答率が 83% である)．実験 3-2 は，CNN，RNN ともに正答率が 70% を下回った．

今回行った実験では，一部で誤り文を判定することがで

表 5 ハッシュ化 2 による CNN での実験 3-1 の結果

番号	学習数	検証数	正答率	再現率	適合率	F 値
1-1	51	23	1.00	1.00	1.00	1.00
1-2	74	33	0.94	1.00	0.89	0.94
1-3	109	47	0.96	0.96	0.96	0.96
1-4	225	97	0.81	0.48	0.88	0.63
1-5	42	19	0.42	0.10	0.33	0.15
1-6	124	54	0.54	0.48	0.58	0.53
1-7	54	24	0.42	0.50	0.43	0.46
1-8	686	294	0.63	0.69	0.55	0.61
1-9	684	294	0.62	0.60	0.54	0.57
2-1	252	109	0.71	0.68	0.73	0.70
3-1	937	402	0.77	0.73	0.71	0.72
3-2	1623	696	0.91	0.88	0.90	0.89

表 6 ハッシュ化 1 による RNN での実験 3-1 の結果

番号	学習数	検証数	正答率	再現率	適合率	F 値
1-1	51	23	0.91	0.83	1.00	0.91
1-1	74	33	0.21	0.20	0.18	0.19
1-3	109	47	0.98	1.00	0.96	0.98
1-4	225	97	0.57	0.04	0.05	0.05
1-5	42	19	0.42	0.38	0.33	0.35
1-6	124	54	0.22	0.23	0.21	0.22
1-7	54	24	0.67	0.69	0.69	0.69
1-8	686	294	0.52	0.36	0.45	0.40
1-9	684	294	0.52	0.46	0.44	0.45
2-1	252	109	0.58	0.58	0.54	0.56
3-1	937	402	0.59	0.48	0.51	0.50
3-2	1623	696	0.69	0.61	0.66	0.64

表 7 ハッシュ化 1 による CNN での実験 3-2 の結果

学習数	検証数	正答率	再現率	適合率	F 値
792	340	0.33	0.33	0.31	0.32

表 8 ハッシュ化 2 による CNN での実験 3-2 の結果

学習数	検証数	正答率	再現率	適合率	F 値
792	340	0.34	0.34	0.33	0.33

表 9 ハッシュ化 1 による RNN での実験 3-2 の結果

学習数	検証数	正答率	再現率	適合率	F 値
792	340	0.44	0.44	0.43	0.44

きなかった．その理由として次のものがあると考えた．

- 実際の文では教師データにふさわしい文が少ない
- 前処理方法が適切でない
- 学習モデルが適切でない

実際の文では，実験 3 で使用したように，ある程度規則性をもって変化している表現は少なく，修正漏れなども含むことがある．よって初回の添削では置換判定された文でも，単語や節において内容の変化をしている文があり，教師データとして望ましくない文が存在した．本稿では 2000 個未満のデータで実験を行ったが，分類ができなかったという結果を考えると教師データ数が少ないことも大きな原因と考えることができる．

実験 3-1 の結果より，事前にルール学習用の教師データを用意して，誤りとされる共通の表現を学習できるようにすることで分類精度が向上することがわかった．反対に，学習用データを別に用意した実験では，分類が十分にできなかったことが確認されているので，学習用データのバリエーションを増やすことや，実際の論文における変化の特徴を観察して学習データに反映するといった処理を行う必要がある．

前処理ではハッシュ化方法を変更しながら実験した．結果として実験 3-1，3-2 ともに活用形情報を用いない方が良い結果を得られたが，これは文の形態素に活用形情報が多く出現しておらず，学習するだけの情報量がなかったからであると推測できる．

学習モデルについては，実験 3 の結果を参照すると CNN の分類器が最もよい結果であったので，CNN モデルを用いて分類を行うべきと言える．パターン 1-7 など，一部の表現では CNN よりも RNN を用いた方が良いスコアを残す場合があった．この結果から，検出したい表現によって向いている分類器が異なると考えた．複数の分類器を組み合わせたアルゴリズムを用いて精度を向上させる方法についての検討は今後の課題である．

5 おわりに

本研究では変更履歴に着目した論文校正支援方法を提案した．論文から要修正となる文を発見して添削作業支援を行うことを目的とした．LCS アルゴリズムを用いて添削前後の対応関係を特定し，それをもとに差分をとることで単語レベル，機械学習を用いて文レベルでの検出ルールを構築する．実際に判定を行い，その結果から文の検出精度を上げるための考察をした．

今後の課題として，分裂や統合の変化をしているサンプルを充実させて検証することが必要である．また，機械学習での文の評価について，教師データをさらに増やし，学習用データセットの作成を行うことが挙げられる．

参考文献

[1] azu：textlint で日本語の文章をチェックする，Web Scratch，入手先〈https://efcl.info/2015/09/10/introduce-textlint/〉(参照 2020-01-08)．

[2] 山田洋志，竹元義美：文章作成履歴を利用した校正支援機能，全国大会講演論文集，Vol.52，No. メディア情報処理，pp.281-282(1996)．

[3] 坂本俊介，須藤崇志，丸山広，中村太一：形態素解析を利用した文章校正方法の提案，研究報告デジタルドキュメント (DD)，Vol.2009-DD-72，No.17，pp.1-6(2009)．

[4] 工藤 拓：MeCab:Yet Another Part-of-Speech and Morphological Analyzer，入手先〈taku910.github.io/mecab/〉(参照 2021-01-08)．