Vol.2018-NL-235 No.5
Vol.2018-SLP-121 No.5
2018/5/13

敵対的生成を利用した校正要否の識別

竹中 誠[1]　柳瀬 利彦[2]　小泉 敦子[2]　江原 遥[3]

概要：文章の校正は一般には人手によるものであり，これを自動化することは人手コスト削減に大きな寄与を果たす．本研究ではソフトウェアのマニュアル文の自動校正を見据え，校正要否の識別性能を向上させる手法を提案する．校正要否の識別において解くべきタスクは，校正対象の文章を入力とした二値分類タスクである．マニュアルにおいて校正前の文は一般には公開されていないため，分類器の教師あり学習で利用するほど大量の校正前の文を入手することは困難である．一方，校正後の文章は大量に存在する．そこで本研究では，校正後の文から擬似的に校正が必要な文を生成し，教師データとして利用する方法を提案する．提案手法では，人手で書かれた文か生成された文かを見分ける分類器を活用する．これにより，人手で書かれた文と近い校正が必要な文を生成する．実験では，提案手法で拡張した教師データを利用することで，要否の識別性能が向上することを確認した．また，本アプローチによる精度向上の可能性を議論した．

Makoto Takenaka[1]　Toshihiko Yanase[2]　Atsuko Koizumi[2]　Yo Ehara[3]

1. はじめに

製品マニュアルの校正では，大きな人的コストがかかっている．計算機によって校正の要否を判定することができれば，人間の校正者が読むべき文章を減らすことができ，コスト削減に繋がる．

このような背景をもとに，本研究では校正の要否の識別について取り組む．すなわち，未校正の文章を入力とし，校正が必要かどうかを判定する二値識別のタスクに取り組む．ここで，校正が必要な文を正例，校正が不要な文を負例とすると，一般的なテキスト分類のタスクとして定式化でき，教師あり学習によって解くことができる．しかしながら，一般に，公開されている製品マニュアルは多数あるが，未校正の文は公開されていないため，正例は少なく，高精度な分類器を学習するには十分でない．

そこで，本研究では校正後の文から擬似的に校正の必要な文を生成することに取り組む．提案手法として，敵対的な分類器を生成器と組み合わせて学習する手法を提案する．敵対的な分類器とは，人間が作成した真の要校正文か，生成器が作成した擬似的な要校正文かを見分ける分類器のことである．

自動校正の先行研究としては，マルチタスクの枠組みでエンコーダ-デコーダによる校正文生成と編集操作の系列ラベリングを同時に解くことで性能を向上させる手法も提案されている [1]．しかしながら，本研究のように校正事例のデータが非常に少ない状況では必ずしも有効ではないことが想定される．

一方で，SNS などにみられる標準的な日本語の表記から逸脱したテキストを再現するデータの生成が斉藤らによって取り組まれている [2]．また，文法誤りを訂正するために，疑似的な誤り文を生成する手法が澤井らに提案されている [3]．これらの手法は，生成器のみを用いており，提案手法のように敵対的な識別器を用いていない．

画像処理の分野では，敵対的な識別器を用いる事例がみられ，Antoniou らは Generative Adversarial Network (GAN) [4] を擬似的な教師データの生成に用いている [5]．

本研究では，GAN を自然言語処理においてデータ生成に活用する予備的な研究として，識別器と生成器とを独立に学習し，最終出力を生成する際に組み合わせた．実験では，擬似的な要校正文によって，教師データを拡張することで識別性能が向上することを確認した．また，提案手法の限界について議論した．

[1] 首都大学東京
Tokyo Metropolitan University
[2] 日立製作所
Hitachi
[3] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology

© 2018 Information Processing Society of Japan

2. マニュアルの校正

本研究は，ソフトウェアの日本語のマニュアルの校正を題材とする．ソフトウェアのマニュアル校正は，複数の技術者によって書かれた原稿を，ひとつの文書として一貫した文体で整える作業である．技術者は，日本語の母語話者であるため，文法的な誤りは少ない．一方で，「事」か「こと」か，「時」か「とき」かなどの日本語としてはどちらも正しいが，文体として統一されているべき点の修正が求められる．また，対象とするソフトウェアの機能と，マニュアルの解説との対応が取れている必要があるため，書かれている情報の修正が行われる．以下では，対象とするマニュアル校正の事例について説明する．

2.1 校正文対データセット

研究対象のデータセットとして，ソフトウェアのマニュアルの校正前の文と校正後の文のペアを用いた．このデータセットの性質を**表 1** に示す．

データセットは 1084 文対からなる．各文を，IPA 辞書を用いた MeCab[*1] により形態素解析したところ，平均トークン長は 30 程度であった．校正の前後で比べると，各文の文字数，トークン数とも平均で 1 ずつ増えている．

表 1 校正文対データセットの性質

指標	校正前	校正後
文の数	1084	1084
平均文長 (文字)	67.1	68.7
最小文長 (文字)	4	4
最大文長 (文字)	714	658
平均文長 (トークン)	32.1	33.1
最小文長 (トークン)	2	2
最大文長 (トークン)	329	300

2.2 変更種別のアノテーション

校正文対データセットにおいてどのような校正が行われたかを調べるため，表 1 の校正前後ペアのうち，1000 ペアに対して，校正前後文の変更種別のアノテーションを付与した．変更種別は，「表記・体裁の変更」，表現の変更」，「情報の削除」，「情報の追加」，「情報の変更」，「対象外」の 6 種別である．

「対象外」は，校正前後の文の言語が異なるなど，校正の対象外であることを意味する．これを除くと変更の種別は，情報量の変化によって，大きくふたつに分けられる．情報量の変化とは，ソフトウェアのユーザの立場で文を読んだとき，受け取る情報が校正の前後で変化しているかを表す．情報量に変化がなければ，「表記・体裁の変更」と「表現の変更」のどちらかになり，情報量が変化していれば，それ以外の選択肢になる．

以下では，各変更種別について例を挙げる．

表記・体裁の変更　単なる文字種の変更や句読点の変更
- **修正前**　問題が無い
- **修正後**　問題がない

表現の変更　用語の変更や言い回しの変更
- **修正前**　システムの指定を見直してから，再度実行してください
- **修正後**　システムの指定を見直したあと，再度実行してください

情報の削除　要校正文から情報の一部を削除
- **修正前**　期間または時刻の指定を見直してください
- **修正後**　時刻の指定を見直してください

情報の追加　要校正文に情報の一部を追加
- **修正前**　必要な容量を確保してください
- **修正後**　メモリーの設定を見直して，必要な容量を確保してください

情報の変更　提示する情報を変更している
- **修正前**　登録に失敗しました．すでにある名前を登録することはできません
- **修正後**　指定した名前はすでに登録されています

日本語母語話者のアノテータにより，校正前後文からの変更箇所の抜き出しと変更種別の付与を行った．変更箇所が複数ある文の場合には，複数の変更種別が付与される．5 名のアノテータに作業を依頼し，3 名以上によって付与された変更種別を採用した．

変更種別の数を**表 2** に示す．最も多い変更種別は「表現の変更」であり，734 ペアが該当した．ついで 378 ペアに「表記・体裁の変更」が見られた．一方で，情報量の変化を伴う変更は最大で 201 件と，情報量の変化を伴わない変更に比べると少なかった．

情報量の変化を伴わない変更であれば，対象とするソフトウェアについての知識などの背景知識がなくとも，言語的な知識だけで校正が可能であると考えられる．本データセットでは変更の約 7 割のペアに言語的な知識だけで対処可能な変更が含まれ，背景知識を用いないアプローチが貢献できる割合は大きい．

表 2 校正文対データセットの変更種別

変更種別	該当ペア数
表記・体裁の変更	378
表現の変更	734
情報の追加	119
情報の削除	201
情報の変更	193
対象外	6

[*1] http://taku910.github.io/mecab/

© 2018 Information Processing Society of Japan

Vol.2018-NL-235 No.5
Vol.2018-SLP-121 No.5
2018/5/13

3. 提案手法

本研究のタスクは，文を入力とし，校正が必要か不要かを出力する二値分類問題である．本研究ではデータセットが小さいことから，分類器としてシンプルなモデルである Bag-of-words を特徴量とした線形 Support Vector Machines (SVM) を用いた．

本研究では，豊富にある校正後文から擬似的な要校正文を生成することを考える．生成される文のバリエーションを増やしつつノイズを減らすために，品詞一致による擬似的な要校正文の生成モデルと，擬似要校正文と真の要校正文の識別モデルとを組み合わせる手法を提案する．

生成のためのモデルの設計では，次の 2 つの性質を考慮する必要がある．

- 生成される文のバリエーションが限られるが，生成された文に誤りが少ないモデル
- 生成される文のバリエーションに富むものの，生成された文に誤りが多いモデル

本研究では，前者として表層文字列の一致に基づいた要校正文の生成モデル，後者として品詞一致に基づいた要校正文の生成モデルを用いた．後者については，生成された文のうち，より人間が作ったものらしい文を選択することで，擬似教師データに含まれるノイズを減らすことができると考えられる．そこで，人の作成した文と機械的に生成した文を見分ける敵対的な識別器を生成モデルに付加する．

擬似教師データを伴う要校正文識別の概要を図 **1** に示す．要校正文と校正済み文とのペアを，要校正文生成学習器に入力し，要校正文生成モデルを学習する．要校正文生成モデルを用いて，校正済み文から擬似的な要校正文を生成する．

要校正文識別器の学習には，もともとの要校正文と校正済み文のペアに加えて，擬似的な要校正文と校正済み文の両方を用いる．

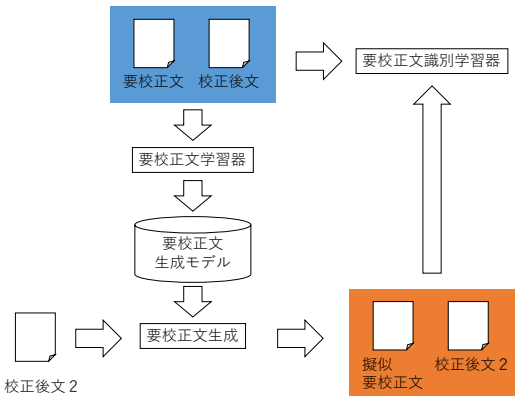

図 **1** 擬似教師データを伴う要校正文識別

次に，敵対的生成器を伴う要校正文識別の概要を図 **2** に示す．図 1 との違いは，擬似的な要校正文を敵対的識別器に入力して選別する点である．敵対的識別器は人間の作った要校正文であるか，それとも要校正文生成器が作った要校正文であるかを見分ける．ここでは，擬似的な要校正文のうち敵対的識別器が最も人間の作ったものらしいと回答した文のみを教師データに追加する．

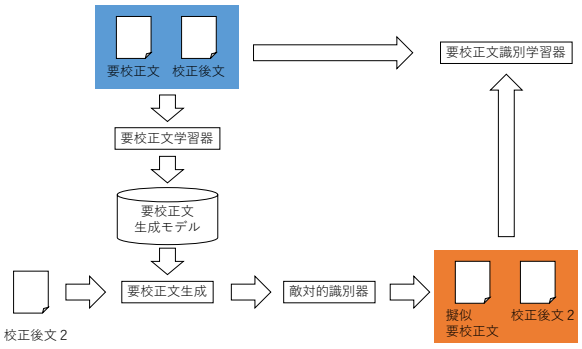

図 **2** 敵対的生成器を伴う要校正文識別の概要

以下，擬似的な要校正文の生成と敵対的識別器のそれぞれについて説明する．

3.1 擬似的な要校正文の生成

校正前後文対 $(s_b, s_a) \in S$ と，対応する要校正文のない校正後文 $s_s \in S_s$ が与えられたとする．擬似的な要校正文の生成では，S から抽出した校正パタン $p \in P$ を文 s_s に適用し，擬似的な要校正文 s_p を生成する．

本節では，擬似要校正文の生成手続きと，表層一致モデルおよび品詞一致モデルについて説明する．

3.1.1 表層一致モデルと品詞一致モデル

擬似要校正文の生成は，(1) パタン抽出と (2) パタン適用の 2 段階からなる．

(1) パタン抽出

本研究では，単純化のため校正前後文対 (s_b, s_a) の Levenshtein 距離が 1 の場合にのみパタンを抽出した．編集距離が 1 の編集操作とは，1 トークンの削除，挿入，置換である．編集操作と編集内容の概要を**表 3** に示す．

表 **3** 編集操作と編集内容 (T は編集操作の対象トークン，B,A は T の前後トークン)

操作	要校正文のトークン列		校正後のトークン列
削除	B T A	$\Longrightarrow$	B A
挿入	B A	$\Longrightarrow$	B T' A
置換	B T A	$\Longrightarrow$	B T' A

編集対象のトークンを T とし，その前後のトークンをそれぞれ B，A とする．削除は，要校正文のトークン列 B T A から T を消し B A とする．挿入は，トークン列 B A の間に T' を追加し B T' A とする．置換はトークン列 B T A の T を T' に置き換え B T' A とする．p は，編集操作名 o と校正前のトークン列 t_b，校正後のトークン列 t_a の組か

© 2018 Information Processing Society of Japan

らなる．例えば，「問題が無い.」を「問題がない.」にする編集の場合，B は「が」A は「.」，T は「無い」，T' は「ない」であり，p は (置換, 「が 無い .」, 「が ない .」) である．

ここで，B と A の一致条件にトークンの表層一致を用いるモデルを表層一致モデル，品詞の一致を用いるモデルを品詞一致モデルと呼ぶ．T, T' はいずれのモデルも表層を使う．

表層一致モデルより品詞一致モデルのほうがマッチ条件が緩和され生成される擬似的な要校正文のバリエーションが多くなることが期待できる．

(2) パタン適用

(1) で抽出した p を s_s に適用し，擬似的な要校正文を生成する．s_s に対して，すべての $p \in P$ について s_s 内のトークン列に t_a が含まれるかを確認し，含まれていた場合には o に従い t_a を t_b に変換する．

例えば，p が (置換, 「が 無い .」, 「が ない .」)，s_s が「反応がない.」のとき，t_a である「が ない .」を t_b の「が無い .」に置換して「反応が無い.」を生成する．

3.1.2 敵対的識別器

前節の手続きで生成した擬似要校正文を敵対的識別器にかけて，人間が生成したものと識別されたものだけを教師データとして選別する．

敵対的識別器の学習と識別の流れを**図 3** に示す．敵対的識別器の学習にも，教師データを利用する．教師データ中の要校正文を正例，校正済み文に要校正文生成器を適用した文を負例として，文の二値識別を解く．

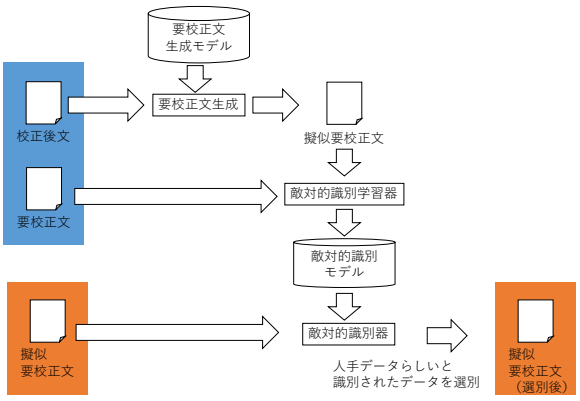

図 3 敵対的識別器の学習と識別の流れ

前節で述べたように，ひとつの校正後文につき，複数の擬似要校正文が生成され得る．敵対的識別器では，そのうち，最も人手で書かれたものらしい文を選ぶ．本研究では，SVM の識別平面からの距離 (decision value) が最も大きい文を擬似要校正文として採用した．ただし，人手による校正後文のラベルが 1，擬似生成された文が -1 とする．

4. 実験

4.1 実験設定

スペース位置のみの修正の場合は分かち書き後の校正前後文が同一文となるため，このような文対は本実験からは除いた．このような文対は 35 文対あり，最終的に実験に用いた文対の数は 1051 文対となった．

10 分割の交差検定を行い，それぞれの手法について要校正文の適合率 (Precision)，再現率 (Recall)，F_1 値の比較を行った．交差検定の 1 試行では，約 900 ペアの教師データと 100 ペアのテストデータでの評価になる．教師データのうち，200 ペアを本タスクの訓練に用いた．残りの約 700 ペアの校正後文を，擬似教師データ生成に用いた．

各文を MeCab を用いて形態素解析し，単語に分割した．特徴量は単語 2-gram に基づく Bag-of-words を用いた．頻度による単語の除外は行わなかった．

線形 SVM のハイパーパラメタは，予備実験によりベースラインの擬似教師データなしの場合の F_1 を最大化するように $C = 0.1$ と定めた．線形 SVM の実装として Scikit-learn [6] を経由して LIBLINEAR [7] を用いた．

4.2 実験結果

実験結果を**表 4** に示す．

ベースラインである擬似教師データなしと比較して，適合率においては表層一致モデルの 2 つの手法が上回り，品詞一致モデルの 2 つの手法は下回っている．再現率においてはこの逆の傾向を示している．

表層一致モデルについては，適合率が 2pt 上昇するが再現率が 7pt 低下と大きく下がる．また，表層一致モデルについては，適合率，再現率ともに敵対的識別器の影響が観られない．

一方，品詞一致モデルについては，敵対的識別器がない場合には適合率の低下が約 13pt と大きいが，敵対的識別器によって擬似教師データを絞り込むことで，約 10pt の適合率の大きな改善が見られる．また，再現率についても約 5pt の改善が見られている．以上のような適合率と再現率の改善の結果，F_1 でみたときには品詞一致モデルに敵対的識別器を組み合わせた手法が最も良い結果となっている．

4.3 生成された擬似要校正文

提案手法によって生成された擬似的な要校正文では，接続詞や助詞の挿入・削除によって情報量が変わらないような擬似要校正文が多く見られた．たとえば，「値を見直してから再実行」という校正後文の文字列から「値を見直して再実行」を生成する事例や，「対象のタスク」から「対象タスク」を生成する事例がみられた．

一方で，品詞一致モデルからは人間らしくない擬似要校正文が多く生成され，敵対的識別器で完全に除去すること

© 2018 Information Processing Society of Japan

表 4　要校正識別の評価結果 (括弧内の数値は標準偏差)

手法	適合率 [%]	再現率 [%]	F_1
擬似教師データなし	68.53 (3.45)	69.46 (3.19)	68.94 (2.77)
表層一致モデル	70.70 (2.83)	62.32 (4.81)	66.11 (3.13)
品詞一致モデル	55.50 (2.06)	74.02 (5.05)	63.34 (2.24)
表層一致モデル + 敵対的識別器	70.67 (2.93)	62.70 (4.94)	66.32 (3.28)
品詞一致モデル + 敵対的識別器	65.18 (2.88)	78.98 (4.64)	71.35 (2.97)

はできなかった．その中でも，特に名詞の前に助詞を挿入するパタンにより助詞が連続する例が多く見られた．例えば「キャプチャを取得」を「キャプチャをの取得」に変換したり，「エラーが発生」を「エラーがを発生」のように変換したりする事例がみられた．

5. 考察

実験結果に基づいて，敵対的な枠組みに適した生成モデルの性質と，本研究での生成モデルの限界について議論する．

5.1 ２つのモデルに対する敵対的識別器の効果

実験では，表層一致モデルに敵対的識別器を加えるよりも，品詞一致モデルに敵対的識別器を加えたほうが，F_1 でみた時に良い結果となっている．これは，予めより強い制約下で生成された文に対して敵対的識別器を入れるよりも，制約を緩めた状態でノイズを許容しつつより多くの文を生成し，その生成した文に対して識別器の性能を向上させるように，識別器の学習に用いるデータを選択するという戦略のほうが最終的な識別器の精度を高め得ることを示唆している．また，品詞一致モデルに対して敵対的識別器を組み合わせることで，適合率を 10pt，再現率を 5pt と大幅に上昇させていることからも，敵対的識別器が識別器の学習にとってノイズとなる事例を除外していることがわかる．

本研究のように教師データが少ない状況においては，一般に高精度な識別器の学習は困難であるが，提案手法の敵対的な枠組みにおいて，識別器の性能を向上させるように擬似的に教師データを拡張させることが可能である．

5.2 提案手法の限界

本研究では，単純化のため生成モデルの教師データを編集距離が 1 の校正に限定している．このため，生成モデルの教師データは 1,051 文対のうち約 15%と少ない．実際の校正では，文字の置き換えや挿入を間欠的に複数個行う場合もあり，そうした校正は提案した生成モデルで再現可能であるため，実際に生成可能な要校正文が 15%というわけではない．それでも，編集が複数トークンにわかれて行われる場合などは本モデルで生成できず，編集距離による生成器の表現能力の限界があることが示唆される．

6. おわりに

本研究では，GAN を自然言語処理においてデータ生成に活用する予備的な研究としてマニュアル文校正の要否を識別する問題に取り組み，校正後の文から擬似的に校正の必要な文を生成する手法を提案した．実験では，擬似要校正文によってデータ拡張することで識別性能が向上することを確認した．

今後の課題としては，生成モデルの表現能力を高めることである．今回は編集距離 1 の校正に限定していたが，より多様な校正に対しても取り組みたい．

参考文献

[1] Yuta Hitomi, Hideaki Tamori, Naoaki Okazaki, and Kentaro Inui.: Proofread Sentence Generation as Multi-Task Learning with Editing Operation Prediction, In Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017)

[2] 斉藤 いつみ, 鈴木 潤, 貞光 九月, 西田 京介, 齋藤 邦子, 松尾 義博: 擬似データの事前学習に基づく encoder-decoder 型日本語崩れ表記正規化, 言語処理学会 第 23 回年次大会発表論文集, 2017.

[3] 澤井 裕一郎, 進藤 裕之, 松本 裕治: 文法誤り訂正のための疑似誤り生成によるラベルなしコーパスの利用, 言語処理学会 第 23 回年次大会 発表論文集, 2017.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio: Generative Adversarial Nets, Advances in Neural Information Processing Systems 27, 2014

[5] Anthreas Antoniou, Amos Storkey and Harrison Edwards: Data Augmentation Generative Adversarial Networks, ICLR 2018 (accepted as workshop paper), 2018

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay: Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011

[7] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang and Chih-Jen Lin: LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research, vol. 9, pp. 1871–1874, 2008

© 2018 Information Processing Society of Japan