オフショア開発向けの事例ベース日本語自動校正システムの構築

鄭育昌　長瀬友樹
株式会社富士通研究所　スピーチ＆ランゲージテクノロジー研究部
{cheng.yuchang, nagase.tomoki}@jp.fujitsu.com

1　はじめに

企業のグローバル化が進むとともに，システム開発などの海外への発注（オフショア開発）が増加している．例えば中国等へのオフショア開発においては，納品物としての技術文書を外国語母語話者が日本語で作成する機会が増えている．しかし，日本語の誤用による技術文書品質の低下が指摘されており，図 1のように，技術文書が納品する前日本人スタッフが人手で文章校正を行なっている．オフショア開発の増加にともない，人間による文章校正コストの増加が問題になっている．このコストを削減するため，すなわち日本人校正者の作業量を削減するため，外国語母語話者のために日本語の誤用を自動的に指摘・修正してくれる「文書校正システム」の開発が期待されている．

オフショア開発に向けた文書校正システムを効率的に構築するためには，まず，我々は，オフショア開発における中国人の執筆した日本語技術文書の校正履歴（表 1）をもとに，中国語母語話者による日本語誤りの傾向を分析した（[鄭ら, 2012]）．我々の校正履歴の分類と分析結果により，最も誤りの頻度が高い助詞変更の校正が最重要課題であることがわかった．しかし，助詞の校正が文の他の部分に影響を受ける場合や語彙誤用など校正方法が文脈に依存する場合，人手で誤りのパターンを一般化して校正ルールを作成することが困難である．そこで，我々は校正履歴をそのままシステムが読み込んで文書校正が動作する事例ベースの校正手法を考案した．事例ベースの校正手法では大量の校正例が必要であるが，我々は実際のオフショア開発会社において業務の中で蓄積された大量の校正履歴を入手することにより，この校正手法を実現できる．

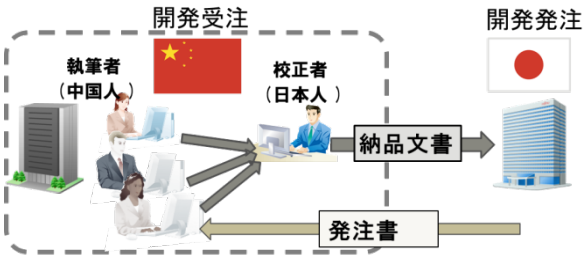

図 1 オフショア開発における技術文書の校正の作業イメージ

本稿で提案した事例ベースの校正システムは，校正履歴コーパスを持ち，1) 事例の検索，2) 事例のチェック，および 3) 対象文の書き直しのステップからなる．

提案手法の校正効果を調べるため，事例ベース手法の再現率に関するシミュレーションを行った．異なる事例数をランダムで選択し，校正ステップに従いテスト事例に対する再現率を測った結果，校正コーパス全体を使用すると，誤用の66%が校正できることが判明した．

2　校正履歴コーパスと事例ベースの校正手法

2.1 校正履歴コーパスの概要

[鄭ら, 2012]で分析された校正履歴コーパスは，オフショア開発の現場で中国語を母語とする技術者が書いた技術文書を日本人が校正したときの作業記録である．表 2 のようなデータを最小単位として含んでいる．校正履歴コーパスの概要と規模は表 1 で示す．

本研究が用いる校正履歴コーパスには一度だけ現れる校正履歴が大量に存在するため，機械学習で実用的な学習結果を得ることが困難である．その原因は，校正履歴を文書執筆者の中国人開発者にフィードバックすることにより，開発者は誤りとその修正方法を学習し，一部の誤りがその後作成した技術文書に現れないことが原因と考えられる．

校正履歴コーパスを効率良く使用するため，本研究は機械学習手法を使用せずに事例ベースの校正手法を提案した．

表 1： 校正履歴コーパスの概要

元文書数	519
執筆者人数	20 人
校正履歴数	8404 件

表 2：校正履歴の例

校正前（負例）	引数*の*エンコード*転換*は*されていない
校正後（正例）	引数*が*エンコード*変換*[DELETE]されていない

2.2 事例ベース校正手法の処理概要

図 2 では事例ベースの校正手法の処理流れを示している．このシステムは，校正履歴コーパスを含んでいる．コーパスの各校正履歴には修正前文字列（誤りを含むもの）と修正後文字列（誤りを修正し

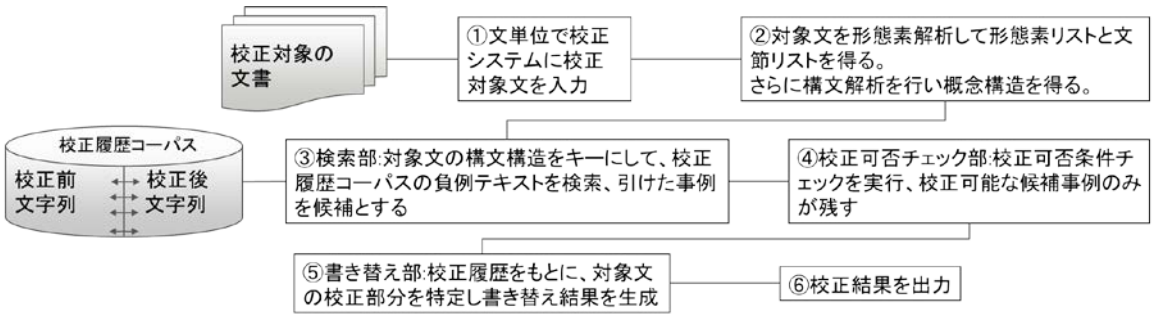

図 2：事例ベース校正システムの処理流れ

たもの）を持つ（表 2）．校正履歴には，誤りを含む校正前の文が負例として，負例を校正した後の文が正例として，対応付けて格納されている．表 2 に示す様に，校正前の文（負例）が「引数*の*エンコード*転換*は*されていない」である場合，この文は，助詞である「の」や「は」の用法の誤り，および語彙「転換」の誤用を含むため，「引数*が*エンコード*変換*されていない」が正例として登録されている．

事例ベースの校正手法は校正履歴コーパスから処理対象文の校正に有用な事例を見つけることにより，幅広い誤り種類の校正および複数誤りの同時校正が可能になる．事例ベースの校正手法の処理ステップは検索部 (図 2 の処理項目③), 校正可否チェック部(図 2 の処理項目④) および書き替え部(図 2 の処理項目⑤)を含む．ステップの詳細な説明を以下にまとめる．

事例ベースの校正処理は，まず校正対象の文書を読み込むと，文書中の文章を文単位に分割し，1 文ずつ，後段の校正処理に出力する(図 2 の処理項目①)．以後の処理に必須の情報を獲得するため，処理対象文に対して，形態素解析，構文解析（係り受け解析）および意味構造解析を行う(図 2 の処理項目②)．これにより，上記形態素解析の結果として，対象文を構成する形態素の一覧である形態素リストと，対象文を文節に分解した文節リストとを取得する．併せて，各形態素の係り元と係り先との構造を含む概念構造を上記構文解析の結果として取得する．以降の処理では，処理項目②で獲得した形態素情報と単語の係り受け情報が必要である．

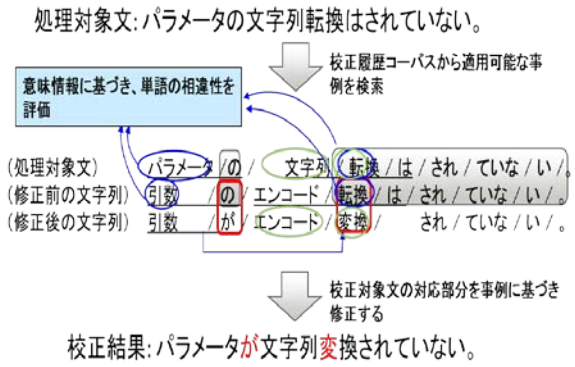

図 3：事例ベース校正処理の一例

図 2 の処理項目③では，校正履歴コーパスから処理対象文の校正に適用可能な事例の候補を検索する．この検索結果は複数の校正事例候補があり，次の処理項目（図 2 の処理項目④）に校正事例候補を出力する．検索結果の事例候補件数が 0 件になる場合，現在の処理対象文を放棄し，次の処理対象文に処理項目②を実行する．

図 2 の処理項目④では，検索結果の校正事例候補に対し，校正事例の校正前（負例）と処理対象文の類似度を計算することにより，校正事例候補の校正可否をチェックする．類似度の計算も処理項目②で獲得した形態素情報と概念構造情報を用いる．類似度の計算値が予め決定したしきい値より下回る場合，事例は候補リストから外し，次の校正事例の類似度を計算する．校正可否のチェックで残した校正事例候補を次の処理（図 2 の処理項目④）に出力する．校正事例候補がこのチェック処理で全部外された場合，処理対象文の校正を放棄し，処理項目②に戻り，次の処理対象文を処理する．

図 2 の処理項目⑤では，校正事例候補を参照し，処理対象文の校正箇所を検出して書き換える．すなわち，各校正事例の校正方法（校正前の文から校正後の文に変換する方法）に因んで処理対象文がの文の校正箇所に対応する部分を書き換えることである．複数の校正事例から書き換えが可能の場合，すべての書き換え結果をユーザに出力する．

次節では，主な処理ステップの処理項目③（検索部），処理項目④（校正可否チェック部）および処理項目⑤（書き換え部）について説明する．

2.3 事例ベース校正手法の詳細処理

図 3 では事例ベースの校正手法による校正処理の一例を示す．校正処理の処理対象文は「パラメータの文字列転換はされていない」 であり，システムはこれに基づき，校正履歴コーパスから適用可能な事例候補：「校正前: 引数のエンコード転換はされていない → 校正後: 引数がエンコード変換されていない」を検索する．その後，システムは校正可否のチェックを行い，事例候補は処理対象文の校正に適用可能だと判断する．その後，システムは校正事例を参照して処理対象文を校正する．すなわち，処理対象文にある助詞「の」を助詞「が」に変更し，処

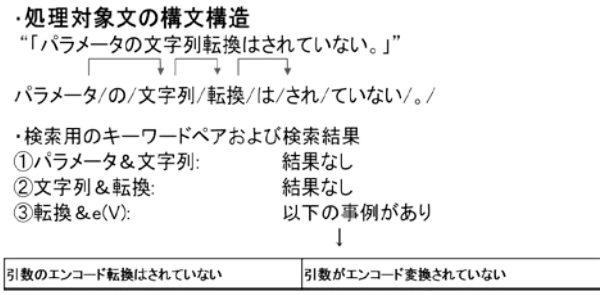

図 4：処理対象文のキーワードおよび校正履歴コーパスの検索結果

理対象文にある語彙「転換」を語彙「変換」に書き換え，更に処理対象文にある助詞「は」を削除する校正である．最後は校正結果の「パラメータが文字列変換されていない」をユーザに表示する．

③検索部: 処理対象文の校正に適用可能な事例候補を校正履歴コーパスに検索する

図 2 の処理項目③の検索部では，処理対象文の係り受け関係を用いて校正履歴コーパスに処理対象文に適用な事例候補を検索する．図 4 では処理対象文「パラメータの文字列転換はされていない」から検索用キーワードを抽出し，校正履歴コーパスの検索結果を示す．システムは処理対象文の中に係り受け関係がある自立語ペアを検索用キーワードとする．

例えば，図 4 のキーワード①では自立語「パラメータ」と「文字列」が係り受け関係があり，検索用キーワードの一つである．キーワードは単語の文字列だけではなく，形態素情報および単語の意味情報も検索用キーワードになる．例えば，図 4 のキーワード③は動詞の「転換」と動詞の語形変化部「e(V)」が検索用キーワードである．検索を行うため，校正履歴コーパスの事例に対し，図 2 の処理項目②で行われる言語処理（形態素解析，構文解析，意味構造解析）を実行する．校正事例に同じ検索用キーワードが存在すれば，適用可能な校正事例の候補リストに入れる．例えば図 4 のキーワードに対し，キーワード③のみヒットした事例がある．

④校正可否チェック部：校正事例は処理対象文に適用できるかどうかをチェックする

図 2 の処理項目④の校正可否チェック部は，校正事例候補リストに残存する校正候補に対し，処理対象文の校正に使用することが可能かどうかのチェックを行う．また，校正可否チェック部は，校正事例候補の絞込みに際し，図 5 のように処理対象文と校正事例（校正前/校正後）とに対する構文解析を実行させる．

具体的には，構文解析を実行した結果，置換文字列が自立語である場合には，置換語と被置換語との間における構文的及び意味的な類似度を評価する．また，置換文字列が助詞または用言の語尾である場合には，該置換文字列を含む文節の自立語同士の類

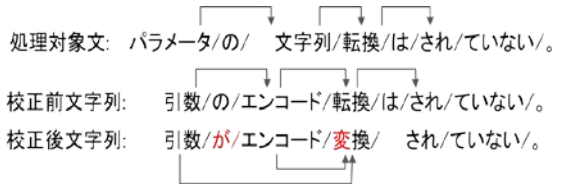

図 5：校正対象文および校正履歴（修正前/後）の形態素解析結果，構文解析結果

似度を評価する．更に，置換文字列が自立語である場合には，該置換文字列を含む文節が係っている単語同士の類似度を評価する．校正可否チェック部は，上記評価により得られた単語間類似度の評価結果に基づき，各々の置換候補に評価点を付与する．そして，この評価点が所定の閾値を下回る置換候補を，上述の校正事例候補から除外する．閾値はシステムの応用先ごとに調整，チューニングすることが可能である．例えば，異なる業種，もしくは異なる会社では閾値の変更により異なる校正事例候補を適用できる．これにより，評価点が閾値以上の校正事例候補のみが，校正候補リストに残ることとなる．

例えば，図 5 の自立語ペア「パラメータ」と「引数」および 自立語ペア「文字列」と 「エンコード」に対し，システムは以下の計算式で自立語の意味一致度の評価点を計算する:

- 一致度 $= \alpha \times \mathrm{Txt} \div \mathrm{WordLen} + \beta \times \mathrm{Sem}$

ここで:

- α, β: 各要素の重み，校正履歴コーパスごとにチューニング可能
- Txt: 単語文字列の編集距離
- WordLen: 単語文字列の長さ
- Sem: 単語の意味クラスの距離

図 5 の事例では，自立語ペア「パラメータ」と「引数」，及び自立語ペア「文字列」と「エンコード」において，今回使用する校正履歴コーパスを提供するオフショア開発会社では同じ使い方をしているため，単語の一致度計算では閾値より高くなり，処理対象文と本事例の負例が類似であることから，本事例は処理対象文の校正に適用可能な事例とする．

次に，適用可能な事例には，校正可能な箇所が存在するかどうかのチェックを行う．校正事例候補に対し，校正前の文と校正後の文における校正箇所（形態素）が処理対象に対応可能であるかどうかをチェックする．例えば，図 5 の事例では校正前の部分形態素列「引数 / の/」が校正後の部分形態素列「引数 / が/」に校正され，校正前の部分形態素列「転換/ は/」が校正後の部分形態素列「変換」に校正されることがわかった．部分形態素列「引数 / の/」は処理対象文の「パラメータ/ の/」に対応し，部分形態素列「転換/ は/」は処理対象文にも存在することから，図 5 の校正事例候補は処理対象文の校正に

適用可能であることが分かった．処理対象文に対応している部分形態素列は校正事例候補に書き換えが行なっていない場合，本事例は処理対象文の校正に使用不能と判断され，校正事例候補リストから除外される．

⑤書き替え部: 処理対象文の校正部分を書き替える

図 2 の処理項目⑤の書き換え部は，現時点で校正事例候補リストに格納されている各校正事例毎の評価点に従い，評価点の高い順に，校正事例候補を並べ替える．次に，最上位に位置することとなった校正事例候補から，校正事例候補の置換文字列により，処理対象文の該当部分（形態素列）を書き換える．ユーザの指定により，複数の校正事例候補による書き換えの結果を表示することが可能である．

図 3 の校正実行例では，自立語「パラメータ」が自立語「引数」と一致（意味的に）であり，かつ形態素列「引数/の/」が「引数 /の/」に校正されたことから，処理対象文の形態素列「パラメータ/の/」が「パラメータ/が/」に書き換えられることが分かる．そのため，処理対象文「パラメータ*の*文字列***転換は***されていない」が「パラメータ*が*文字列***変換***されていない」に校正され，校正結果としてユーザに提示できる．

3　事例ベース校正手法の効果のシミュレーション

前節で説明した事例ベース校正手法には，類似度を計算するための係数および閾値を予め調整，チューニングする必要がある．係数の調整は行なっているため，本稿では，事例ベース校正手法の正解率を正確に計算することができない．事例ベース校正手法の評価は代わりに本手法を用いる最大効果を図るシミュレーションを行い，システムの最大再現率（recall）を推測した．

具体的には，我々の校正履歴コーパスから一部の事例をテストデータとして抽出し（324 事例），残りの校正履歴を図 2 の校正履歴コーパスとし（8080 事例），テストデータに対する校正箇所指摘の再現率を図る．図 2 の処理項目④の類似度計算の際，必須な閾値と評価関数の係数をきめていないため，類似度の判断は人手で判断する．すなわち，単語が一致（類似）するかどうかは人間の判断によるものである．

テストデータも校正履歴であるため，各履歴には表 2 のように校正前と校正後の文がある．シミュレーションでは各校正前の文（負例）に対して校正処理を行う．校正履歴コーパスにテストデータを正しく校正できる事例が存在すれば，校正可能と判断する．しかし，システムの適合率（precision）をシミュレーションするには，すべての校正履歴コーパスがテストデータに適用可能（不正な校正）な場合を検出する必要がある．人手による類似度の評価回数が膨大になり，シミュレーション工数も膨大化になる．そのため，本稿では再現率のシミュレーションのみを行った．

シミュレーションでは，校正履歴コーパスの規模と校正効果の関連性を検証するため，異なる規模の校正履歴コーパスを用いて検証を行った．具体的には，ランダムで全校正履歴コーパスから一定件数の履歴のみを取り出して上記の検証を行う．今回のシミュレーションは 4 つの規模で 5 回の検証を行った．表 3 では 5 回のシミュレーション結果を示す（Sim1 から Sim5 まで）．たとえば，Sim 1 の検証では，ランダムで校正履歴コーパスから 1000, 3000, 5000 の事例を抽出し，各コーパス規模における校正効果（Recall）を検証する結果である．Sim2 以降は同じ方式でランダムに 1000, 3000, 5000 件の事例を抽出し，検証を行った．最大件数（8080 件）を用いる場合，テストデータに対する校正の再現率は 65.8%であることがわかった．

表 3 :事例ベース校正手法のシミュレーション結果

Corpus size	Sim 1	Sim 2	Sim 3	Sim 4	Sim 5
0	0.0%	0.0%	0.0%	0.0%	0.0%
1000	33.3%	32.7%	30.1%	34.2%	36.3%
3000	49.7%	53.2%	47.1%	55.8%	52.9%
5000	57.6%	58.5%	57.3%	62.3%	59.9%
8080	65.8%	65.8%	65.8%	65.8%	65.8%

4　まとめ

本稿では，校正履歴をそのままシステムが読み込んで文書校正が動作する事例ベースの校正手法を説明した．事例ベースの校正システムは校正履歴コーパスを持つ．校正処理は，1) 事例の検索部，2) 事例の校正可否チェック部，および 3) 対象文の書き換え部のステップからなる．

まず，処理対象文の単語依存構造をキーとし，校正履歴コーパスに同じ単語依存構造を持つ事例を検索する．その後，検索結果の適用候補事例に対し，事例が処理対象文の校正に適用できるかどうかを確認する．この場合は，処理対象文と校正事例（修正前の文）との共通部分が事例中の校正部分に似ているかどうかを確認する．その後は，チェックされた適用事例候補を用いて，校正事例の修正方法と同様に処理対象文を校正する．事例ベースの校正手法の最大効果を調べるため，異なる事例数をランダムで選択し，校正ステップに従いテスト事例に対する再現率を測るシミュレーションの結果，校正コーパス全体を使用すると，誤用の 66%が校正できることが判明した．

参考文献

[1] 鄭育昌, 長瀬友樹: 外国語母語話者が作成する日本語技術文書を対象とした校正履歴の分析, 第18回言語処理学会年次大会, pp.34-37 (2012).