

文書ファイルを対象にした授業課題における 事前確認システムの開発

宮崎 章太^{1,a)} 伊藤 匡祐¹ 島袋 舞子¹ 兼宗 進¹

概要：本稿では Microsoft Office Word で作成された文書ファイルを自動正誤判定するシステムについて報告する。本システムを使うことで、学生は授業課題として作成した文書ファイルを Web ページからアップロードし、教員があらかじめ登録した正解文書ファイルとの相違点を判定結果として受け取ることができる。ファイルの比較を行うために、Word の保存形式である docx ファイルから XML ファイルを抽出し、比較を行うことで提出ファイルと正解ファイルの差分を検出している。

キーワード：文書作成, Microsoft Word, 授業支援システム, 自動正誤判定

Development of a System for Pre-check Assignments for Document Files

MIYAZAKI SHOTA^{1,a)} ITO KYOSUKE¹ SHIMABUKU MAIKO¹ KANEMUNE SUSUMU¹

1. はじめに

情報社会の発展に伴い大学を始めとした教育機関ではコンピュータを利用した文書作成方法について学ぶ授業が実施されている。多くの場合、文書作成ソフトとして Microsoft Office Word *¹が利用されている。

このような授業では学生が授業の課題に沿って Microsoft Office Word で作成した文書ファイルは現状では教員が手作業で採点する必要がある。しかしこの方法では学生が提出する大量のファイルを手作業で確認する必要があるため、教員の負担となっている。また、学生の文書作成技術向上のためには、「教員が文書ファイルを確認した後、修正箇所を示し、学生が修正する」ことを適切な設定が行われるまで繰り返すことが効果的だが、教員への負荷が高く学生の作成した課題を繰り返し確認することは難しいことから、学生の学習の足かせとなってしまっている。これらを

解決する 1 つの方法として、正しい文書ファイルと学生が作成したファイルと比較し、正誤判定を行うことが考えられる。

Microsoft Office Word は 2 つの文書ファイルと比較する機能を有しているが、これを授業課題の正誤判定として利用するには学生が予め模範解答の文書ファイルを保持している必要があり、これでは授業課題の正誤判定ツールとして使用することはできない。そこで自動で文書ファイルを正誤判定するシステムがあれば、学生は自分のタイミングで何度でも誤り箇所の確認と修正を繰り返して課題に取り組むことができると考えた。

これまでに Microsoft Office ツールの正誤判定には、VBA を用いたツール [1] や生成された Office Open XML を解析することで採点するシステム [2][3][4] が存在する。学習者が採点と点数の把握に利用できるものの、これらのツールやシステムは設問の設定など、採点に細かい設定が必要となる他、学習者に対して誤っている箇所を示す機能は見当たらない。また、Microsoft Office の操作スキルの測定を目的とし開発されたシステムも存在する [5][6]。正誤判定により誤り箇所を示すが、これらはシステムに用意された

¹ 大阪電気通信大学
Osaka Electro-Communication University

^{a)} eh21a081@oecu.jp

^{*1} Microsoft, Microsoft Office Word, Microsoft Office Word 2007 および Word は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

問題のみに対応している。

そこで、教員によって予め模範解答ファイルを設定しておき、学生は自分の作成した文書ファイルをアップロードするだけで自動正誤判定を行うことができるシステムを開発することにした。このシステムにより、授業を受講する学生は課題として作成した文書ファイルを手軽に誤り箇所を確認、修正して課題に取り組むことができる。

また、本システムの特徴として課題の追加、設定が簡単であることが挙げられる。類似の研究やシステムでは課題の作成、設定するために専用の定義ファイルを用意する必要があるものが多く、そのために Word や正誤判定システムへの深い理解が必要である。本システムでは模範解答の文書ファイルと簡単な設定のみで課題を追加することができる。

2. システムの設計

本システムはコンピュータを利用した文書作成方法について学ぶ際に Microsoft Office Word を利用する授業を支援する目的で設計した。授業を受講する学生が実習課題として作成した文書ファイルを自動正誤判定する Web アプリケーションである。

2.1 採点の流れ

本システムでは学生が提出した文書ファイルと教員が予め登録した模範解答の文書ファイルを比較し、その差分を検出することで正誤判定を行う。

Microsoft Office Word 2007 以降のバージョンではファイルの保存形式として docx ファイルが使われている。docx ファイルの実態は ZIP 圧縮された XML ファイルや画像ファイルである。解凍後の docx ファイルのディレクトリ構造例を図 1 に示す。

これを調査すると document.xml には Word 本文の基本的な要素が示されており、header*.xml や footer*.xml にはそれぞれヘッダーとフッターの内容が記述されていた。また、styles.xml にはアウトラインなどのスタイルが記述されていることがわかった。これらのファイルの内容を解析し、文書ファイルのテキストなどの要素を取得し、それらを比較することで文書ファイルの差分を取得することができると思われる。document.xml の内容例を図 2 に示す。

採点ツールとしては、予め模範解答の文書ファイルを登録し、それと学生が提出したファイルを前述の方法で比較し、結果を学生に示すことで実現できると考えた。

2.1.1 XML ファイルの整形

docx ファイルの XML には ID のような要素や、1 行のテキストが複数の XML 要素に分けられてしまう、編集過程で発生した不要な要素が残るといったある種の断片化が含まれるため、単純にファイルを比較することはできない。そこで、XML ファイルを比較する前に XML ファイ

ルから不要な要素を取り除いたり、断片化を修復することで XML ファイル同士の比較精度を向上させることができると考えられる。

本システムでは不要な XML タグ、XML タグに付与された不要な属性の削除と同一段落内のテキストのうちフォントやフォントサイズなどの装飾要素が等しいものを結合するといった処理を行っている。

2.1.2 XML ファイルの比較

XML ファイルの比較には、Python の difflib モジュールを使用した。Linux 等の diff コマンドに相当する行単位の比較を行うことができる。検出した XML ファイル間の差分から、その行がどのような設定を保持しているかを取得し、その内容をユーザに表示する。

2.2 フォントの比較について

本システムでは「フォント」「テキスト」のように項目ごとに採点の有効無効を切り替える機能を実装した。フォントについては、現在は「MS ゴシック」のような具体的なフォントについて比較を行っている。しかし、課題によっては「ゴシック体」のような指定を行うことも考えられるため、今後はフォントのグループ化や、他の項目についても比較の幅を調整できるような機能を検討したい。

3. 実装

本システムは Python を使用して実装を行った。Web サーバには nginx を使用し、Python の Web アプリケーションフレームワークに Flask を使用した。

3.1 アプリケーション概要

本システムは学生が簡単に使用できるように Web アプリケーションとして実装した。

図 3 に、システム構成図を示す。学生は正誤判定実行ページを使い、作成した課題を確認する。教員は管理ページを使い、課題の登録を行う。

3.1.1 正誤判定実行ページ

図 4 に、正誤判定実行ページの画面を示す。学生はこのページからシステムを利用する。正誤判定実行ページでは正誤判定を行う授業課題と正誤判定を行いたい文書ファイルを選択し、文書ファイルの正誤判定を行うことができる。

3.1.2 管理ページ

図 5 に、教員用の管理ページを示す。管理ページからは、課題名、模範解答の文書ファイル、正誤判定の対象要素を設定することができる。正誤判定の対象要素では「テキスト」「フォント」などの、検出する内容を選択する。

3.2 正誤判定機能の実装

3.2.1 XML ファイルの整形

XML ファイルの正誤判定では以下の処理を施している。

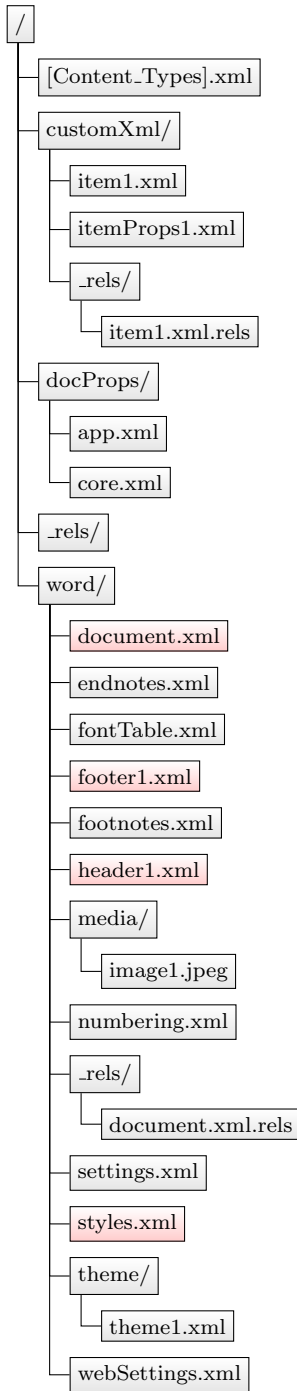


図 1 docx ファイルの構造例

- 不要なタグの削除
- 不要な属性の削除
- 装飾要素が同じテキストを結合
- style.xml の内容を document.xml に挿入

XML ファイルの整形では、XML の構文を解析して木構造の Python オブジェクトに変換し、独自に定義した条件に従ってタグやタグの属性を削除を行っている。条件の例として、以下に「属性が一つもない”w:rFonts”を削除する」や「”w:p”タグについて”id”の属性を削除する」を説明する。

- 「属性が一つもない”w:rFonts”を削除する」ではフォ

```

1 <w:p w14:paraId="3738411F" w14:textId="2
  B29362A" w:rsidR="004E7C76" w:rsidRPr="003
  A4544"
  w:rsidRDefault="00B118CF" w:rsidP="00
  D26F26">
2
3 <w:pPr>
4   <w:widowControl />
5   <w:jc w:val="left" />
6 </w:pPr>
7 <w:r>
8   <w:rPr>
9     <w:rFonts w:hint="eastAsia" />
10  </w:rPr>
11  <w:t>あいうえお</w:t>
12 </w:r>
</w:p>

```

図 2 document.xml の内容例

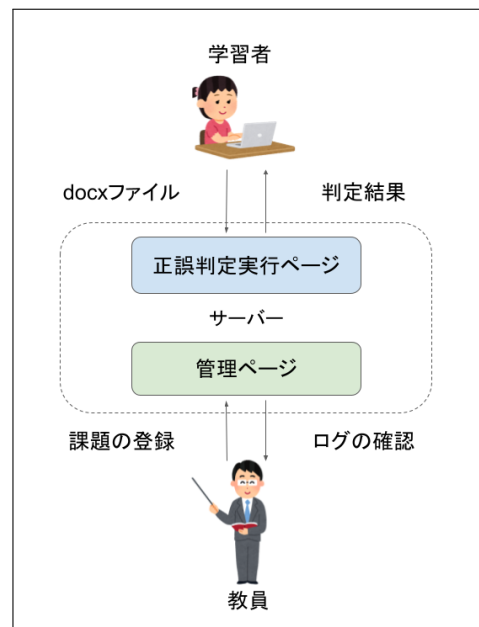


図 3 システム構成図



図 4 正誤判定実行画面

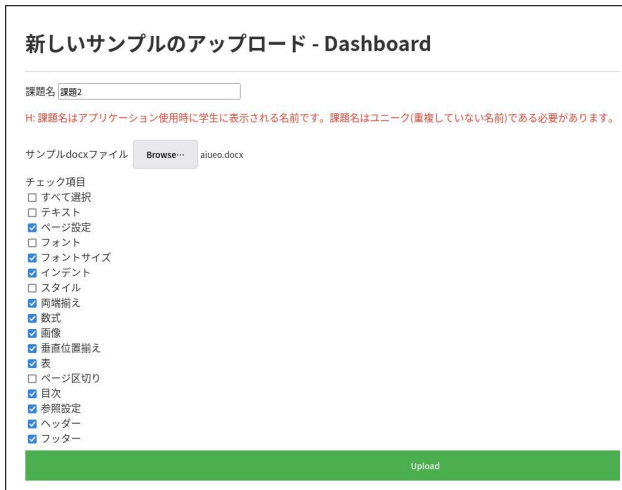


図 5 教員用管理ページ

ント設定を持たないフォント定義タグを削除対象としている。docx ファイルでは編集手順等により事実上情報を持たない定義タグが生成される。このようなタグは比較の際に誤検知の原因となるため、削除する必要がある。

- 「w:p」タグについて「id」の属性を削除する」ではタグに付与されている ID を持つ属性を削除対象としている。XML 上に存在する ID 情報の多くは、Word ドキュメントの見た目とは関係がなく、比較の際に誤検知の原因となるため、削除する必要がある。

テキストの結合ではフォントやフォントサイズといった装飾が等しい連続したテキスト同士を結合している。Word では編集手順により、1 行のテキストが複数の XML 要素に分けられてしまうことがある。断片化した XML の例を図 6 に示す。図 2 と図 6 はどちらも「あいうえお」と 1 行に書かれたテキストの XML であるが、図 6 では「あいう」と「えお」が別の XML 要素として構成されていることがわかる。さらに付随するテキスト装飾要素である「rFonts」タグの属性を見てみると、どちらも同じフォントが設定されており、テキストの装飾情報は同じであることがわかる。

このようにテキストが分割されてしまった状態はファイルを比較する際に障害となるため、文字列を結合する前処理を行うようにした。本システムでは、同一段落内 (XML 上では 1 つの「w:p」タグに含まれる範囲) で連続しており、かつ装飾設定が等しいテキストを結合して一つの「w:t」タグで表すことができるように処理を適用している。

Word 上でのアウトラインのような設定では適用されているスタイルが重要となる。document.xml 上にはスタイルの情報は記述されておらず、図 1 に示す style.xml 上に記述されている。そしてそれらを紐付ける ID が document.xml 上に記述されている。スタイルの ID はファイルによって異なるため、document.xml 上のスタイル ID が記載されているタグ内に紐付けられたスタイル情報を直接挿入するこ

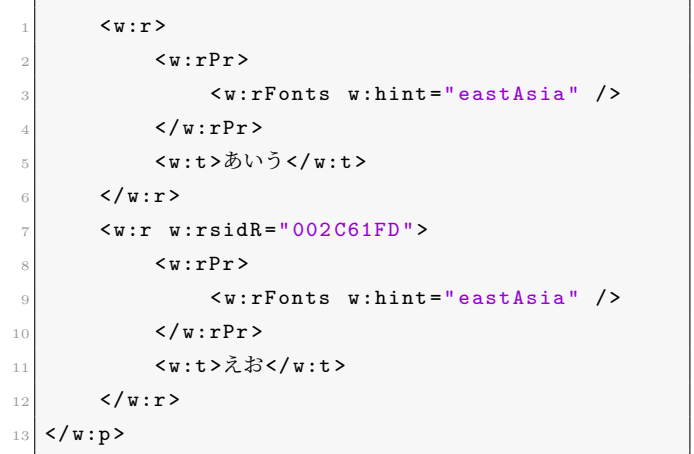


図 6 断片化した XML の例

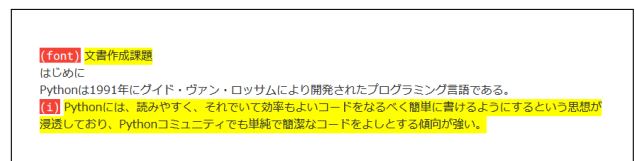


図 7 訂正箇所への提示例

とでファイル同士を比較可能にしている。

3.2.2 整形した XML ファイルの比較

本システムは前項で整形した XML ファイルのテキストを Python の difflib を使って比較している。diff により出力された差分内容の行を XML として解析し、そのタグ名からその行が持つ情報を判断している。例えば出力された行のタグが「rFonts」であればフォントの設定が間違っているという旨のメッセージを正誤判定結果としてユーザーに表示する。

3.2.3 誤り箇所の提示

誤り箇所をユーザーに示す際、行数など場所を特定できる情報が必要となる。しかし Word の XML 上には行数などの情報が含まれていない。そこで、本システムでは文書ファイル上のテキストをすべて抽出し、そのテキストの訂正箇所にマーカーをつけることでユーザーに誤り箇所を示している。ユーザーへの誤り箇所の提示例を図 7 に示す。図 7 に示すように文書ファイル内のテキストが表示され、誤りがある箇所が黄色くハイライトされる。誤り箇所の初めに赤くハイライトされたタグが表示される。タグにより「font」（フォントの設定ミス）や「i」（インデントの設定ミス）といった誤り内容を示している。

4. おわりに

本稿では Microsoft Office Word の自動正誤判定システムの開発について報告した。本システムでは授業受講者は Web ブラウザからサイトにアクセスし、課題を選択し、文書ファイルをアップロードすることで正誤判定結果を受けることができる。また、教員は模範解答の文書ファイルを

アップロードすることで課題を登録することができる。このシステムにより、教員の負担軽減と受講者の学習品質の向上が期待できる。

今後は授業において本システムの実用を行うなどして、有用性の調査を行いたい。

参考文献

- [1] 渡邊光太郎: Microsoft Excel による試験採点システムの製作, 城西大学情報科学研究センター 城西情報科学研究, Vol.18, No.1, pp.15-24 (2008).
- [2] 松井吉光, 岩田員典: 情報リテラシーのための統合システム構築と運用- Microsoft Excel・Word 自動採点システムを含む-, 愛知大学情報メディアセンター紀要「COM」, Vol.31, No.1, pp.1-10 (2022).
- [3] 岩田員典: Office Open XML 形式の Excel (®) ファイル自動採点システム, 愛知大学情報メディアセンター紀要「COM」, Vol.29, No.1, pp.1-15 (2019).
- [4] Li, H., and Qingguo, G.: The Design of Automatic Scoring System for Document Operational Test Question of MS-Office, Proceedings of the 2014 International Conference on Education, Management and Computing Technology, pp.393-396 (2014).
- [5] 高木正則, 瀬戸山光宏: MS-Office 操作スキルを測定するコンピュータ適応型テストの開発と評価, 情報処理学会 情報教育シンポジウム論文集, Vol.2019, pp.220-225 (2018).
- [6] 高木正則, 瀬戸山光宏: コンピュータ適応型テストを利用した大学1年生の MS-Office 操作スキルの実態調査, 情報処理学会 情報教育シンポジウム論文集, Vol.2018, pp.215-220 (2018).