

HITsにおけるWord文書の採点プログラム2020年度版の開発

松 井 吉 光
谷 口 正 明

要 旨：本学名古屋校舎で開講される一般教養科目「情報リテラシー・入門，応用」では本学で独自に開発したe-learningシステムであるHITsが使われている。本稿は2020年度にHITsのWord部門で使用する予定の文書解析・採点プログラムwt2020.pyの開発について述べる。wt2020.pyにおける主な変更点は，エラーメッセージの表示に関する部分である。

キーワード：e-Learning，教材，自動採点，Word[®]，Python

1 はじめに

キーボード入力，文書作成および，表計算は学生にとって大学教育において，また卒業後の就業時において，必須のスキルであろう。本学名古屋校舎ではそういった基本的なスキルを学生に身につけてもらうために2006年度から「情報リテラシー・入門，応用」を開講して来た。この講義は対面教育とe-Learningを併用していて，e-Learning部分で用いているのがHITs¹である。HITsとはタイピング練習とMicrosoft社のWord[®]とExcel[®]の練習問題を対話的自動採点する，本学で独自に開発したe-Learningシステムである[3]。筆者グループは，これまで継続的にHITsのWord[®]部門の解析・採点プログラムを改良してきた[4]–[8]。本稿では，Word部門の採点プログラムであるPython3で記述されたwt2020.py

の開発について述べる。

現行のwt2019.py[8]は2019年度の「情報リテラシー・入門，応用」で利用し，正解・不正解の判定について大きな支障は報告されていない。しかし，学生のアンケートでは不正解時のメッセージのわかりにくさが指摘されている。そのため，wt2020.pyでは，学生の間違いをエラーメッセージに反映させる部分の改善を試みた。以降wt2019.pyからwt2020.pyへの変更点について述べる。

2 wt2019.pyのエラーメッセージの不親切な点

wt2019.pyのエラーメッセージがどのように不親切かということ，例えば，

そして軽く跳りあがる心を制しながら，そ

¹ Highly Interactive Training system

² 梶井基次郎「檸檬」より

(2)

HITsにおけるWord文書の採点プログラム2020年度版の開発

の城壁の頂きに恐る恐る檸檬を据えつけた。²

という段落において、「恐る恐る檸檬」を太字にしなさいという問題があったとしよう。この問題の採点制御ファイルは

```
##第1問
```

```
##恐る恐る檸檬
```

```
##「恐る恐る檸檬」の部分を太字に設定する
```

```
1 b
```

```
1 27 33
```

```
太字
```

```
;;
```

となる。解答者が最初に

そして軽く跳りあがる心を制しながら、その城壁の頂きに恐る恐る檸檬を据えつけた。

のように間違っ「頂に恐る恐る檸檬」を太字にする設定だけをしたとする（以降、不正解例1呼ぶ）。このファイルをwt2019.pyで採点すると、「変更箇所が見つからないか、変更範囲が間違っています」というエラーメッセージが表示される。これのどこに問題があるかという、「変更箇所が見つからない」と、「変更範囲がまちがっている」ことを区別せずに不正解を出している点にある。不正解であることは間違いないが、エラーメッセージを分かりやすく表示するという点ではいささか不親切な仕様となってしまう。

では、どうしてwt2019.pyがそのような採点結果になるかという、その原因は採点のプロセスにある。間違っ解答例の採点を例に、wt2019.pyの採点プロセスを具体的にしてみる。wt2019.pyでは、まず最初に採点対象の段落の構造を分析し、どのようにrun要

素に分割されているかを見る。先の解答の例では、

1. そして軽く跳りあがる心を制しながら、その城壁の (1, 24)
2. 頂きに恐る恐る檸檬 (24, 33) (太字)
3. を据えつけた。(33, 40)

という3個のrun要素が発生している。括弧内の左の数字は段落の最初から何文字目かを表し、右の数字は左の数字にそのrun要素の文字数を足した数字になっている。次に採点制御ファイルの「1 27 33」のうちの2番目の数字が上の括弧の左の数字を比較して一致するものがあるか調べ、3番目の数字が上の括弧の右の数字を比較して一致するものがあるか調べる。上の不正解の解答例では上の括弧の左の数字に「27」がないので、その段階で不正解と判断し、それ以上調べずにエラーメッセージを生成する。そのため、「変更箇所が見つからない」と「変更範囲がまちがっている」は区別しない仕様になっている。ここで期待されるメッセージは、「変更範囲がまちがっている」のみである。

ちなみに同じように間違っ解答ファイルであっても、期待される「変更範囲がまちがっている」または、より正確な「指定の範囲を越えて変更されています」をエラーメッセージを表示する場合がある。それは、runの要素が

1. そして軽く跳りあがる心を制しながら、その城壁の (1, 24)
2. 頂きに (24, 27) (太字)
3. 恐る恐る檸檬 (27, 33) (太字)
4. を据えつけた。(33, 40)

のように分割される場合である。こうしたことが起こる原因としては、あえて「頂に」と「恐る恐る檸檬」を別々に選択して太字にし

た、または、「頂に」を含むその場所以前の範囲、または「恐る恐る檸檬」を含むその場所以降の範囲に別の修飾（例えば斜体字にする）を加えた場合が考えられる。この場合は、括弧内の左の数字の中に「27」があり、括弧内の右の数字の中に「33」があるため、wt2019.pyはこの段階では不正解を出さずに、次のチェック項目に進む。次に「恐る恐る檸檬」の文字列が太字の修飾がなされていることを確認して、この解答例では太字になっているのでさらに次のチェックを行う。その次のチェックは、対象の前後のrun要素に不必要に太字の修飾が為されているかであり、この不正解の解答例ではその前のrun要素「頂き」が誤って太字になっているため、不正解と判断する。このとき生成されるエラーメッセージが「指定の範囲を越えて変更されています」となるのである。

また、別の不正解の例を考えてみよう（以降不正解例2と呼ぶ）。解答者が

そして軽く跳りあがる心を制しながら、その城壁の頂きに恐る恐る檸檬を据えつけた。

のように間違って指定の範囲の一部「恐る檸檬」だけを太字にする設定したとする。この場合run要素は、

1. そして軽く跳りあがる心を制しながら、その城壁の頂きに恐る (1, 29)
2. 恐る檸檬 (29, 33) (太字)
3. を据えつけた。(33, 40)

のように分割される。これをwt2019.pyで採点した場合、エラーメッセージは「変更箇所が見つからないか、変更範囲が間違っています」となる。これは先述の例と同様、括弧の右の数の中に採点制御ファイルの「1 27 33」のうちの2番目の数字「27」に一致する数が

ないために、その段階で不正解と判断され、エラーメッセージが生成されるからである。ここで表示されるべきエラーメッセージとしては、「変更範囲がまちがっている」であろう。

今回、wt2020.pyでの改善の目標は、解答者がどのような編集プロセスを経ても

そして軽く跳りあがる心を制しながら、その城壁の頂きに恐る恐る檸檬を据えつけた。

という解答があったら、「指定の範囲を越えて変更されています」というエラーメッセージが生成されることである。また、指定の範囲の一部が正しく変更が為されていれば、「変更範囲がまちがっています」が表示されるようにすることである。

3 wt2020.pyにおける変更点

wt2019.pyでは正解・不正解の判定に不具合が見つかっていないので、wt2020.pyで行った変更点としては主に不正解を判断した後にエラーを調べるプロセスを追加したことである。具体的にその変更点について、再び前節の問題を例にして述べる。

追加したプロセスでは、採点対象の段落の中にrun要素に採点対象箇所「恐る恐る檸檬(27,33)」を含むrun要素すべてを見つける。括弧の右側の数字のなかで、採点対象範囲の開始位置「27」を超える数字を前から順番に探し、見つけたらその一つ前のrun要素を区間最初のrun要素とする。次に括弧の左側の数字の中で、採点対象範囲の終了位置「33」以上の数字を前から順番に探し、見つければそれを区間最後のrun要素とする。その後、区間の最初から最後までrun要素を調べ、いずれかのrun要素に指定の修飾がなされていれば、エラーメッセージとして「指定の変更がなされていますが、変更範囲が間違っていま

(4) HITsにおけるWord文書の採点プログラム2020年度版の開発

す」というメッセージを表示し、そうでなければ「指定の変更が該当箇所になされていません」というメッセージを表示するようにした。

不正解例1の場合、区間のrun要素は

2. 頂きに恐る恐る檸檬 (24, 33) (太字)

の一つであり、それに太字の修飾がなされているので、「指定の修飾はなされているが、範囲が間違っています」というメッセージが表示される。不正解例2の場合、区間のrun要素は

1. そして軽く跳りあがる心を制しながら、その城壁の頂きに恐る (1, 29)
2. 恐る檸檬 (29, 33) (太字)

の二つであり、2つの目run要素に太字の修飾がなされているので、同様に「指定の修飾はなされているが、範囲が間違っています」というメッセージが表示される。他にも解答ファイルが

そして軽く跳りあがる心を制しながら、その城壁の頂きに恐る**恐る檸檬**を据えつけた。

のような修飾がなされていて、run要素が

1. そして軽く跳りあがる心を制しながら、その城壁の (1, 24)
2. 頂き (24, 26)
3. に恐る (26, 29)
4. 恐る檸檬を (29, 34) (太字)
5. 据えつけた。(34, 40)

のようになっていた場合、区間のrun要素は

3. に恐る (26, 29)

4. 恐る檸檬を (29, 34) (太字)

となる。この場合も太字に修飾されたrun要素があるため「指定の修飾はなされているが、範囲が間違っています」というメッセージが表示される。

以上が、wt2020.pyの主な変更点である。

4 まとめ

本稿では、wt2019.pyからwt2020.pyへの主な変更点について述べた。wt2019.pyまでのバージョンではそれ以外に、エラーメッセージそのものがわかりにくいところがあったので、もう少し親切なメッセージへの変更を行った。今後は学生の意見も聞きながら、より親切なエラーメッセージになるよう改善していきたい。同時に、制御ファイルの自動作成にも取り組んでいく予定である。

参考文献

- [1] Standard ECMA-376 Office Open XML File Formats, <http://www.ecma-international.org/flat/publications/standards/Ecma-376.htm>
- [2] XML Path Language バージョン1.0 W3C 勧告, <http://www.w3.org/TR/1999/REC-xpath-19991116>
- [3] “Excel, Word自動採点システムHITsの構築と運用” 岩田貞典, 他, 愛知大学情報メディアセンター (20)No.1(2010)
- [4] “HITsにおけるWord文書の採点プログラムの開発” 長谷部勝也, 松井吉光, 谷口正明 愛知大学一般教育論集 (40) 25-40 (2011)
- [5] “HITsにおけるWord文書の採点プログラム2013年度版の開発” 長谷部勝也, 松井吉光, 谷口正明 愛知大学一般教育論集, (45) 41-53 (2013)
- [6] “HITsにおけるWord文書の採点プログラム2016年度版の開発” 松井吉光, 長谷部勝也, 谷口正明 愛知大学一般教育論集, (52) 27-36 (2017)
- [7] “HITsにおけるWord文書の採点プログラム2018年度版の開発” 松井吉光, 谷口正明 愛知大学一般教育論集, (54) 43-49 (2018)

- [8] “HITsにおけるWord文書の採点プログラム
2019年度版の開発”
長谷部勝也, 松井吉光, 谷口正明
愛知大学一般教育論集, (56) 31-39 (2019)

