

moodleQmaker

Sumiyo Nishiguchi[†] Ikuma Ishizawa[‡]School of Management, Tokyo University of Science[†]Graduate School of Management, Tokyo University of Science[‡]

1. 多肢選択問題作成プログラムの概要

Moodle には問題作成に用いる高性能なインターフェイスがある。マニュアルを読むことなく手軽に問題を作成することが可能となっている。しかし、一問作成するに当たり雑多な手順の踏まなくてはならない点が問題である。

この問題を解決するために、GIFT 形式のテキストファイルを作成しインポートする方法をとることで高速かつ多数の問題を作成することができる。GIFT 形式とは特定のタグで文章を囲み問題を作成する方法のことである。

今回は TOEIC でよく出題される問題形式の多肢選択問題、とりわけ動詞の活用に変化した問題を作成するプログラムを作成した。

このプログラムは、プログラムに与えた文から動詞を探しだし、動詞を空欄にしたものを問題文とし、選択肢には抜き出した動詞を正解選択肢、動詞の変化形を間違い選択肢として GIFT 形式に形成し出力する。一文に動詞が複数存在する場合はランダムに選択される。動詞の活用語尾、接尾辞の選択肢は Natural Language Tool Kit(NLTK)によって生成した。プログラムの必要条件としては、プログラムは Python3.3 を用いて記述し、動詞の検索には NLTK を用いた。また生成した動詞の活用が正しいか判断するために Web 英和辞典と合わせる機能を含んだ。この機能は Beautiful Soup4 を用いた。

2. TOEIC リーディング問題の接辞選択肢の作成

TOEIC のリーディング問題には、形態素の選択問題が多い。接辞の知識があれば、選択肢の単語の意味を知らなくても正解を選ぶことは往々にして可能である。問題例 1 は空所の前後関係から名詞を選択すればよいと推論し、名詞の接尾辞は *-ence* であるという知識がもしあれば、正答可能である。問題例 2 は、動詞の屈折接尾辞も選択する問題である。

[問題例 1]

The _____ location for our new office is downtown.¹

- (A) preference
- (B) preferably
- (C) preferable
- (D) prefer

[問題例 2]

Beginning the first of next month, lunch breaks will be _____ by fifteen minutes.²

- (A) short
- (B) shortened
- (C) shortening
- (D) shortage

このような、動詞の接尾辞に関する問題は、語幹（と接頭辞）と接尾辞との組合せ問題であるため、選択肢の自動生成が可能である。音声また音韻等の要因による変化があるため、修正が必要である場合も往々にしてあるが、かなりの程度予測可能である。

(1) pre-fer + {ence, abl(e) + y, able, ø}

(2) short + { ø, en + ed, en + ing, age}

ならば、語幹と接尾辞との組み合わせを考えれば、TOEIC の問題作成は可能である。形態素解析をし、語幹を抽出し、接尾辞との組み合わせを 4 種類作ればよい。よって、NLTK により、英文から動詞を抽出し、語幹を取り出し、正答以外に他の屈折また派生接尾辞を付けた語を 3 つ生成することにした。

3. 問題作成手順

1. Input.txt 内に以下の形式に沿って作成したい問題数分を入力する。

[問題形式]<TAB>[総選択肢数]<TAB>[問題文]

問題形式は T: 多肢選択問題

M: 丸バツ問題

J: 記述 をサポートしている。

2. moodleQmaker を実行する。

3. Output フォルダ内の問題数分の*.txt が出力される。

4. プログラムの流れ

moodleQmaker 内で働く処理は大きく分けて三つの処理グループに分かれる。入出力グループ、問題作成グループと web 問い合わせグループである。

4.1. 入出力グループ

入出力グループは moodleQmaker.py 本体に書かれている。入力グループには関数として main():moodleQmaker の本体がある。以下の関数は main から呼び出される。

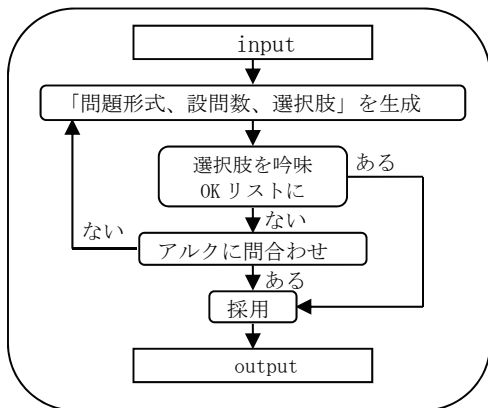
input1(): input.txt から読み込むための関数。

Qt(): 多肢選択型問題の作成用関数。

Qj(): 記述問題の作成用関数。

¹ Lin Lougheed, *Longman Preparation Series for the TOEIC Test, Introduction Course*, 5th Edition, pp122, Pearson, NY, 2007.

² Lin Lougheed, *Longman Preparation Series for the New TOEIC Test, Intermediate Course*, 4th Edition, pp313, Pearson, NY, 2012.



Qm() : ○×問題の作成用関数。

Qwrite() : OUTPUT フォルダへの出力用関数。

4.2. 問題作成グループ

問題作成グループは外部関数として lang.py に書かれている。問題作成グループには、以下の関数が存在する。

cut() : Qt,Qj から呼び出される。NLTK を用い問題文を分解し品詞タグをつける関数。

tamonn() : cut 関数から呼び出される多肢選択問題の問題文作成関数。動詞リストも作成する。

ran_sel() : tamonn 関数から呼び出される。動詞リストから、どの動詞を正解選択肢にするかランダム抽出する。

setumonn() : tamonn 関数から呼び出される。正解選択肢から間違い選択肢を作成する。

seikai() : setumonn 関数から呼び出される。正解選択肢を GIFT 形式に加工する。

matigai() : setumonn 関数から呼び出される。間違い選択肢を GIFT 形式に加工する。

stem() : setumonn 関数から呼び出される。間違い選択肢を作成するために正解選択肢の原形を活用させる。

kijyutu() : 記述問題用回答例を作成する関数。

1. Input.txt 内の指定の形式を一行ずつ

[問題形式]を itemList[0]

[総選択肢数]を itemList[1]

[問題文]を itemList[2] として読み込む。

2. itemList[0]と itemList[1]を引数に input 関数を呼び出し、問題名と出力ファイルの接頭字を決める。

3. 問題形式が判明すると、Qt、Qm、Qj、Qa、Qk から形式にあった関数を選択する。

ここからは多肢選択問題を例に取り上げていく。

":::問題名:::問題文{~誤~誤=正~誤}

":::" + [2 で決定した問題名、ここでは “多肢選択”]+ [input.txt の何行目か、何問目か。ここでは “1”]+ ":::" +lang.cut(src,input2,qtype2)

4. lang.cut()は問題作成グループ内の関数である。NLTK を利用して問題文を分解、タグ付を行う。今回探す品詞は動詞なので、タグが"VB" "VBD" "VBG" "VBN" "VBP" "VBZ"の単語を探す。探した

単語をリスト形式でまとめる。

5. 問題種類別の問題作成関数に 4.で作成したリストを引き渡す。Tamonn()関数を使い問題文を作成する。リスト中に複数の正解選択肢になりえる候補がある場合がある。この場合、ran_sel 関数により候補からランダムに選び出す。正解選択肢とする単語が決定すると、問題文中の単語を [----]と置き換える。ここで問題文は完成である。

6. Setumonn()関数を利用し[----]と置き換えた単語を元に、誤り選択肢を作成する。置き換えた単語は活用している場合があるので nltk.PorterStemmer().stem()を使い原形に変形させる。正解選択肢は seikai 関数で処理して GIFT 形式の正解選択肢型に変形する。設問数分、誤り選択肢を作成する。総設問数－正解選択肢(1問)問作る。誤り選択肢は suffix をランダムで選び、stem()で原形+suffix を作成する。ここで、生成した語が OK.txt に存在している場合そのまま採用。ここで web 問い合わせグループに処理をわたす。

4.3. web 問い合わせグループ

OK.txt にも NG.txt にも生成語が存在しない場合は Web 英和辞典に問い合わせる。問い合わせした結果存在する場合、その語を OK.txt に追加し採用する。問い合わせの結果、存在する語ではない場合は NG.txt にその語、原形、ランダム値を書き込み、もう一度ランダムで suffix を選ぶ。

また選択肢の重複が起こらないようにするために、作成した選択肢を NG リストに保存しておく。それぞれの誤り選択肢は matigai 関数で処理して GIFT 形式の間違い選択肢型に変形する。

このリストから、選択肢をシャッフルして、正解選択肢の位置を毎回変わるようにする。出力された文字列である選択肢全体を{}でくくり GIFT 形式に形成する。

7. Tamonn 関数と Setumonn 関数の出力を文字列として結合させる。この文字列は 3.の lang.cut()の戻り値として返される。

ここで問題作成グループの仕事は終了する。

8. Output フォルダ内に出力する。

これで作成は終了である。

5. 結論

本研究では、TOEIC リーディング問題に接尾辞の選択に関する問題が多いことに着目し、Moodle にインポートできるファイル形式で問題の自動生成プログラムを開発した。NLTK を使用して Wikipedia 等の英文より動詞を抽出し、語幹を切り離して接尾辞と合成して間違い選択肢を作成した。開発したプログラムは自身の辞書データを構築していくため、web に負荷をかけることなく問題精度と速度を向上させることができる。接頭辞に関しても同様の方法が可能であるし、動詞以外の品詞、例えば名詞を抽出しても同様の問題作成が可能である。