Test for CZ1003 Assignment 2

by Liu Siyuan from FS1

Introduction

This is the file to show my test results for my program. You will see various examples of my program down below. In my test, my program displays the time used to autocorrect the word. In the program, there are two function to enable / disable this feature.

Interface

When the program is started, it will display a welcome screen to show you the usage of this program:

```
**************************************************
Hi, you are using my autocorrection system!

======================USAGE=======================
1. Use autocorrect("word") to autocorrect the word
2. Use add_word("word") to add your own words
3. Use add_extra_words() to extend the word list
4. Use remove_extra_words() to remove extra words
5. Use display_time() to display time consumption
6. Use not_display_time() to disable time display
==================================================

Please be notified that it may take a slightly
longer time for some words to autocorrect after
extra words are added

The program took 1.583 sec to set it up
**************************************************
```

Examples for autocorrection

The program will only show at most 6 results, with words with lowest word distance and highest word frequency in front.

```
>>> autocorrect("flyy")
Did you mean: fly, or flay?
Most likely word is: fly
It took 0.120 sec to autocorrect
>>> autocorrect("pxcture")
Did you mean: picture, posture, pictures, pictured, pasture, or lecture?
Most likely word is: picture
It took 0.127 sec to autocorrect
>>> autocorrect("bunting")
Your spelling is correct.
It took 0.007 sec to autocorrect
>>> autocorrect("thx")
Did you mean: the, tax, th, or thy?
Most likely word is: the
It took 0.067 sec to autocorrect
>>> autocorrect("tcip")
Did you mean: trip, or tip?
Most likely word is: trip
It took 0.129 sec to autocorrect
>>> autocorrect("abut")
Did you mean: but, or about?
Most likely word is: but
It took 0.114 sec to autocorrect
>>> autocorrect("sherlcok")
Did you mean: sherlock?
It took 0.183 sec to autocorrect
>>> autocorrect("errror")
Did you mean: error, horror, terror, mirror, or rrrr?
Most likely word is: error
It took 0.125 sec to autocorrect
>>> autocorrect("diatance")
Did you mean: distance, distances, or pittance?
Most likely word is: distance
It took 0.185 sec to autocorrect
>>> autocorrect("publica")
Did you mean: public, publican, publicly, or publish?
Most likely word is: public
It took 0.138 sec to autocorrect
```

Examples for addtional features

1. Add unrecognized word

First, the program doesn't recognize the word "omg". After I added it to the word list,

the program recognizes it.

```
>>> autocorrect("omg")
Did you mean: ogg?
It took 0.067 sec to autocorrect
>>> add_word("omg")
>>> autocorrect("omg")
Your spelling is correct.
It took 0.045 sec to autocorrect
```

2. Allow words that are contractions

The program can recognize contrations, and autocorrect contraction with contractions

as results

```
>>> autocorrect("cnn't")
Did you mean: can't, don't, cent, won't, isn't, or cannot?
Most likely word is: can't
It took 0.128 sec to autocorrect
>>> autocorrect("don't")
Your spelling is correct.
It took 0.081 sec to autocorrect
```

3. Add extra words from "word_list.txt"

The word "wongshy" is not in the "big.txt" but in the "word_list.txt". At first, the

program cannot find any similar results. After I added extra words from "word_list.txt",

the program shows that the spelling is correct.

```
>>> autocorrect("wongshy")
No similar word found.
It took 0.158 sec to autocorrect
>>> add_extra_words()
Extra words are added to the word list
>>> autocorrect("wongshy")
Your spelling is correct.
It took 0.161 sec to autocorrect
```

4. Indicate the most probable word

If there are more than one results for autocorrection, the program prints one more line indicating the most probable word.

```
>>> autocorrect("comerphenxion")
Did you mean: comprehension?
It took 0.776 sec to autocorrect
>>> autocorrect("hella")
Did you mean: hell, hello, halls, heels, wells, or lelya?
Most likely word is: hell
It took 0.146 sec to autocorrect
```

Extra testing

To fully test my program, for both bugs and speed, I wrote another python program which store all the words in a article to a list and iterate through the list to test the speed and whether there is any error. The article was modified by me so that most of the words inside that article is wrong. You can find the article *here* if you are interested.

Below is the result showing how many words are tested and the minimum, average and maximum time used to autocorrect.

```
1689 words have been tested
min 0.000938, avg 0.292654, max 1.871369
```