Course: CS562
Name: Yanshen Wang
Date: 02/29/16
Bug/Progress Report

1. **Describe some bugs I found**
   In this term of CS562 class, my project is to use the TSTL tool to test a python library and find out whether or not there exist some bugs in those programs. In fact, this is a really interesting library because it is used for translating the pronunciation of Chinese characters into Latin alphabet. As we know, in currently, it is very difficult for the programmers to design a perfect application which can convert each of Chinese characters to the Latin alphabets accurately. Therefore, after I try lots of times to put different Chinese characters into this library, I successfully find some bugs.

   a) According to the syntax of the Chinese characters, under the different situations, one character may have two or more pronunciations. However, after I tested this python library in several times, I found that most Chinese characters only have one pronunciations, and absolutely this causes lots of bugs. For example:

      p.get_pinyin(u"开辟") -->u'kai-pi'

      p.get_pinyin(u"复辟") -->u'fu-pi'   correct pronunciation: fu-bi

      p.get_pinyin(u"扁担") -->u'bian-dan'

      p.get_pinyin(u"扁舟") -->u'bian-zhou   correct pronunciation: pian-zhou

      p.get_pinyin(u"禅让") -->u'shan-rang'

      p.get_pinyin(u"禅师") -->u'shan-shi'   correct pronunciation: chan-shi

   b) Furthermore, when I tried to pick up some uncommon Chinese characters and put them into the library, I found that even though this program can successfully recognize most of Chinese characters, there still exists some characters which cannot be recognized. For example:

      p.get_pinyin(u"巭") -->u'\u5ded'   correct pronunciation:bu

      p.get_pinyin(u"嫑") -->u'bao'   correct pronunciation:biao

      p.get_pinyin(u"砉然") -->u'huo-ran'   correct pronunciation:xu-ran

2. Explain progress to date

Actually, the whole progress of this testing project is based on my original proposal, and I implement them in the following steps:

a. First of all, in the syntax of Chinese pronunciation, some characters can have two or more pronunciations. However, after I put those kind of Chinese characters into this library, I found that this program cannot consider this situation, which means each Chinese character only has one pronunciation.

b. In addition, I will try to input some mixed strings into the function to find out if this library will meet some bugs. Actually, this library did really well in this situation since it can smartly recognize which strings are Chinese characters and which strings are not. For the Chinese characters, it can convert them into Latin alphabet; while for the other strings, it just keep their original type.

c. Furthermore, I want to test the maximum length of input strings in those functions, and for example in the ***get_pinyin*** function, I will input a long Chinese poem to detect whether or not this python program will meet any problem. In fact, in this situation, no matter how many Chinese characters I put into this library, it can translate them successfully.

d. Usually, the Chinese characters include two kinds of different types: one is called simplified Chinese characters, and the other is called Traditional Chinese characters. When I test those two types of Chinese characters with this library, I found that it can recognize both of them very well.

e. Finally, I will try to pick up some uncommon used Chinese characters to test if this program can recognize those characters accurately. Actually, in this library most Chinese characters can be translated successfully, but there still exits some characters which are not recognized.

3. Estimate my progress by end of term

So far, I just find 2 kinds of bugs, which includes the multiple Chinese pronunciations issue and the uncommon used Chinese characters issue. Therefore, by the end of term, I hope I can find more bugs from this library.

4. Quality of the SUT

Personally, I think this is a good library, but not a very perfect library since there still exists some bugs to be fixed. For the multiple Chinese pronunciations issue, I think this is a big problem and the designers did not consider well about this aspect of problems when they build this library. I know it is very difficult to solve this problem, but it absolutely causes lots of errors. For the used Chinese characters issue, I think this is not a very critical problem because most common Chinese characters can be recognized successfully.

5. Code coverage

After I used the “timeout” command to test my code coverage, the result shows that it is 64.95%, and maybe in the next step I will do other things to improve my code coverage.