Proposal for testing

Student Name: Yipeng Wang
Date: Jan, 24, 2016

For this testing project, I found a third party python library, which called ftfy. The reason I choose this library is this library is very interesting and useful in our daily lives. The meaning of ftfy is fixes text for you and the goal of ftfy is to take in bad Unicode and output good Unicode. It's not used to take in non-Unicode and output the Unicode. In addition, it also cannot protect users from write Unicode-aware code. Sometimes, we will meet this kind of situation: our input is decoded properly and no errors, but we cannot get the correct output which we want. Ftfy is used to fix this problem, which could be found at https://github.com/LuminosoInsight/python-ftfy.

Ftfy is used to fix Unicode which broken in various ways. It works in Python 2.7, Python 3.2 or later edition. The most interesting kind of brokenness is that ftfy will fix the different standard between encoded Unicode and decoded. Usually, it will output some nonsense characters which is called "Mojibake". "ftfy.fix_text()" is the main function of ftfy, which will fix various different problems. For example, if the text contain HTML entities like: &amp, which replace some certain characters, we can use it to get what the characters actually are. In addition, "ftfy.explain_unicode()" could show what's going on in the string, which used to debug. "ftyfy.fixes.decode_escapes()" could show the built-in "Unicode_escape" codes does, but this one will not cause mojibake. "ftfy" cannot fix all of the mix-ups, but it could understand the text which is decoded as: Latin-1, Windows-1252, Windows-1251, MacRoman and cp437. Since, it states "ftfy" could be used in various single-bytes encodings, I think it is good enough for the project.

To test this project, all of single-bytes encodings which "ftfy" is supported should be tested is necessary. In addition, since "ftfy" works on Python 2.7, Python 3.2 and later edition, I also need to test all of the functions to check whether it works on each Python edition. Moreover, I will test different single-byte encodings which are supported by "ftfy". Since "ftfy" also has some function calls; such as: "uncurl_quotes" could preserve quotation marks, but sometimes the mojibake will also generate some quotation marks which will bring some glitches, I will test it to check whether the quotation marks which comes from decoding will lose too much accuracy for "ftfy". On the other hand, I will also test after fix the mojibake by "ftfy", whether it will change the space between each word, which means whether we can get a sequence of proper readable characters. For some mathematic characters, I will test whether it will lose the superscripts or subscripts after fixing the mojibake by "ftfy". "ftfy" could support actual CJK, so I will check whether adding the actual CJK characters in mojibake strings will cause some problems.

For future work, as a TSTL beginner, I will do my best to finish this test project for "ftfy" library in past of this term. Since, "ftfy" cannot handle the non-UTF encodings used for Chinese, Japanese and Korean, if I have much more time, I will think about how fix that if possible.