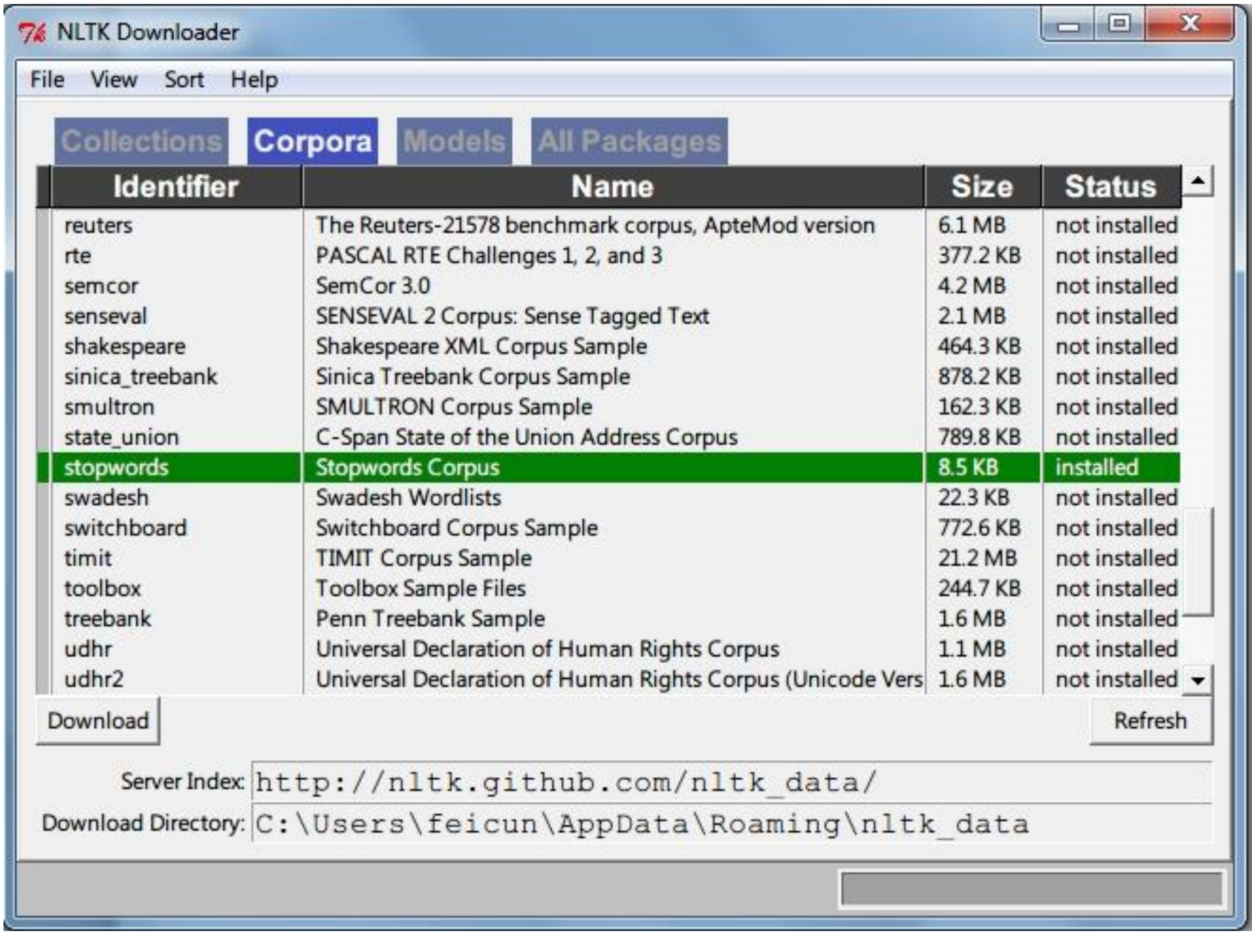
First of all, I'm sorry to make my program too complicated; you need to modify many places before start to test my program. I tried my best to avoid confusing, but the program still looks ugly…

Here is the instruction of how to modify the program:

1. Install nltk from: http://nltk.org/install.html
   I used nltk to checking stop words
   I encountered several problems when I was installing nltk on my Windows 7, so if you encounter some problems too, this link might helpful:
   http://www.fbagirov.com/2012/10/13/installing-nltk-for-python/

2. Install stopwords Corpus for nltk:
   a. Open Python command line and run: >>> import nltk
   b. Run: >>> nltk.download()
   c. In the nltk downloader, click "Corpora" from the top
   d. Scroll down, click "stopwords" and install it.
   e. As the below picture show:



| Identifier | Name | Size | Status |
|---|---|---|---|
| reuters | The Reuters-21578 benchmark corpus, ApteMod version | 6.1 MB | not installed |
| rte | PASCAL RTE Challenges 1, 2, and 3 | 377.2 KB | not installed |
| semcor | SemCor 3.0 | 4.2 MB | not installed |
| senseval | SENSEVAL 2 Corpus: Sense Tagged Text | 2.1 MB | not installed |
| shakespeare | Shakespeare XML Corpus Sample | 464.3 KB | not installed |
| sinica_treebank | Sinica Treebank Corpus Sample | 878.2 KB | not installed |
| smultron | SMULTRON Corpus Sample | 162.3 KB | not installed |
| state_union | C-Span State of the Union Address Corpus | 789.8 KB | not installed |
| stopwords | Stopwords Corpus | 8.5 KB | installed |
| swadesh | Swadesh Wordlists | 22.3 KB | not installed |
| switchboard | Switchboard Corpus Sample | 772.6 KB | not installed |
| timit | TIMIT Corpus Sample | 21.2 MB | not installed |
| toolbox | Toolbox Sample Files | 244.7 KB | not installed |
| treebank | Penn Treebank Sample | 1.6 MB | not installed |
| udhr | Universal Declaration of Human Rights Corpus | 1.1 MB | not installed |
| udhr2 | Universal Declaration of Human Rights Corpus (Unicode Vers | 1.6 MB | not installed |

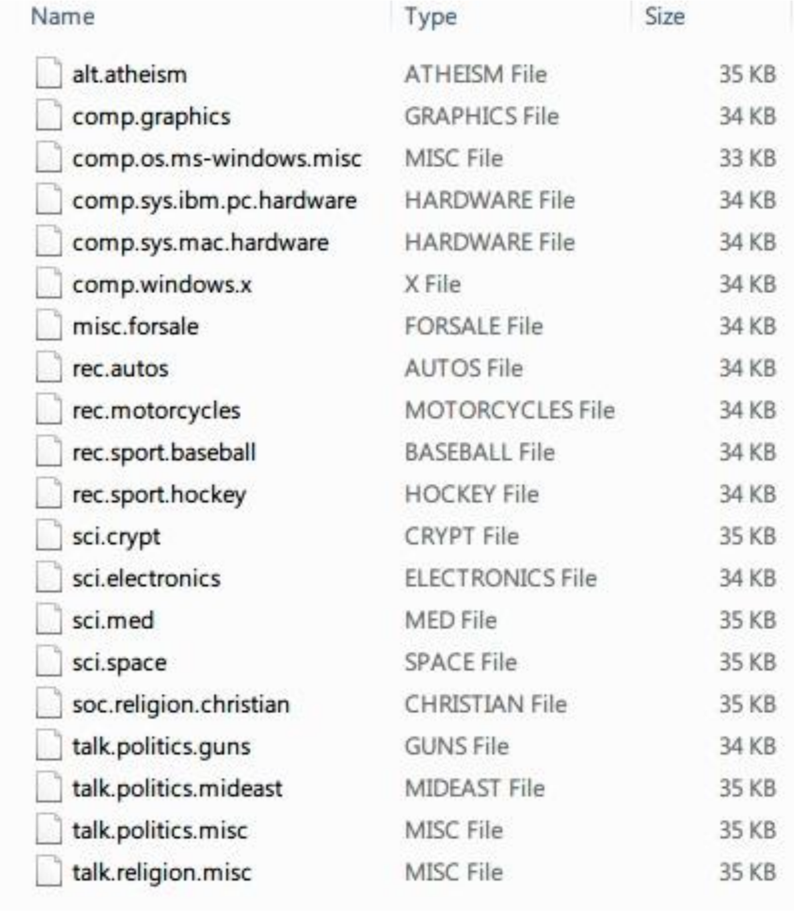
Server Index: http://nltk.github.com/nltk_data/

Download Directory: C:\Users\feicun\AppData\Roaming\nltk_data

3. Modify the directory paths from **tfidf.py** file:
    a. In line 147 there has: path = 'C:/Users/feicun/hw5/20_newsgroups'
    Please modify this path to where you put the 20_newsgroups directory. The next level of 20_newsgroups directory should be the newsgroups directory.
    b. In line 161 there has: path = 'C:/Users/feicun/hw5/TFIDFCategoryFiles'
    Please create a directory named "TFIDFCategoryFiles" under hw5, and modify the path in this line. We will need copy the pickle files of category TFIDF dict to this directory.
    c. In line 179, there has path = 'C:/Users/feicun/hw5/TFIDFCategoryFiles'
    Please modify this path to directory "TFIDFCategoryFiles"

4. Modify the directory paths from **tester.py** file:
    a. In line 13, there has : path = 'C:/Users/feicun/hw5/20_newsgroups'
    Please modify this path to where you put the 20_newsgroups directory. The next level of 20_newsgroups directory should be the newsgroups directory.

5. Run "generateTFIDFCategoryFiles()" function before you start to test my program.
    - You can call this function from the main method of **tfidf.py** file.
    - This function will generate all 20 category TFIDF dicts, and write them to pickle files, so you can load them directly without calculate the dicts again and again.
    - This function needs very long running time. In my 4 years old laptop, it takes around 50 minutes.
    - The pickle files are named by each newsgroup, like the following picture shows, they MUST be put in the "TFIDFCategoryFiles" folder.



| Name | Type | Size |
|---|---|---|
| alt.atheism | ATHEISM File | 35 KB |
| comp.graphics | GRAPHICS File | 34 KB |
| comp.os.ms-windows.misc | MISC File | 33 KB |
| comp.sys.ibm.pc.hardware | HARDWARE File | 34 KB |
| comp.sys.mac.hardware | HARDWARE File | 34 KB |
| comp.windows.x | X File | 34 KB |
| misc.forsale | FORSALE File | 34 KB |
| rec.autos | AUTOS File | 34 KB |
| rec.motorcycles | MOTORCYCLES File | 34 KB |
| rec.sport.baseball | BASEBALL File | 34 KB |
| rec.sport.hockey | HOCKEY File | 34 KB |
| sci.crypt | CRYPT File | 35 KB |
| sci.electronics | ELECTRONICS File | 34 KB |
| sci.med | MED File | 35 KB |
| sci.space | SPACE File | 35 KB |
| soc.religion.christian | CHRISTIAN File | 35 KB |
| talk.politics.guns | GUNS File | 34 KB |
| talk.politics.mideast | MIDEAST File | 35 KB |
| talk.politics.misc | MISC File | 35 KB |
| talk.religion.misc | MISC File | 35 KB |

6. Now you can start to test my program, the tester.py will display the result in cmd. And you can call “hCluster()” function from the main method of **tfidf.py** file.