Information Retrieval Lab 1

1. Objectives

- Try the Jieba Chinese segmenter
- Try NLTK English tokeniser and Porter stemmer
- Compute frequency distribution of Chinese / English text

2. Jieba

Basic Test

Start python:

```
import jieba

seg_list = jieba.cut("我来到北京清华大学", cut_all=True)

print("Full Mode: " + "/ ".join(seg_list))  # 全模式
```

Different Options

Find the difference between the following options:

```
seg_list = jieba.cut("我来到北京清华大学", cut_all=True)
print("Full Mode: " + "/ ".join(seg_list))  # 全模式

seg_list = jieba.cut("我来到北京清华大学", cut_all=False)
print("Default Mode: " + "/ ".join(seg_list))  # 精确模式

seg_list = jieba.cut("他来到了网易杭研大厦")  # 默认是精确模式
print(", ".join(seg_list))

seg_list = jieba.cut_for_search("小明硕士毕业于中国科学院计算所，后在日本京都大学深造")  # 搜索引擎模式
print(", ".join(seg_list))
```

Accuracy of Jieba in Tokenising Text

In Python:

```
import jieba

para = '西北大学信息科学与技术学院成立于2005年5月，是由前计算机科学系和电子科学系为主体整合而成。其中，计算机科学与技术专业于1971年设立，计算机科学系于1981年成立；半导体物理与器件和无线电电子学专业于1958年设立，电子科学系于1992年成立。'

t = jieba.cut( para )
tokens = [ x for x in t ]
# list of strings

print( tokens )
```

How accurate is the result? Here is a longer text to try:

```
para2 = '西北大学信息科学与技术学院成立于2005年5月，是由前计算机科学系和电子科
学系为主体整合而成。其中，计算机科学与技术专业于1971年设立，计算机科学系于1981年
成立；半导体物理与器件和无线电电子学专业于1958年设立，电子科学系于1992年成立。信
息学科是国家“211工程”重点建设学科。近年来承担了国家“973”、“863”、国家自然
科学基金等多项科研项目；获得了国家科技进步二等奖等30多项国家和省部级奖励，相关成
果被 bbc、《泰晤士报》及中央电视台《科技博览》、《走进科学》栏目、《光明日报》和
《陕西日报》等多家新闻媒体报道。承担了国家级、省级教学研究项目多项；获得了国家教学
成果二等奖3项；出版教材30余部，包括国家规划教材5部；建设国家精品资源共享课程2门，
国家双语课程1门，省级精品资源共享课程7门。'
```

3. NLTK English Tokeniser

To make the tokeniser work, you need some files on your C drive. Check for this file: c:\nltk_data\tokenizers\punkt\czech.pickle . If it does not exist,

1. Create c:\nltk_data if it does not exist.
2. Inside nltk_data, create c:\nltk_data\tokenizers
3. From Moodle, get punkt.zip
4. Copy the punkt directory from the .zip to c:\nltk_data\tokenizers so that you now have c:\nltk_data\tokenizers\punkt\czech.pickle and many other similar files.

In Python:

```
import nltk

tokens = nltk.word_tokenize( 'this is a sentence.' )

# Now try with a longer text.

para3 = 'The Northwest University School of Information Science and
Technology was established in May 2005. It is a combination of the
former Department of Computer Science and the Department of
Electronic Science. Among them, the computer science and technology
major was established in 1971, the computer science department was
established in 1981; the semiconductor physics and devices and radio
electronics major was established in 1958, and the electronic
science department was established in 1992. The information science
is the key construction discipline of the national "211 Project". In
recent years, he has undertaken many scientific research projects
such as the national "973", "863", and the National Natural Science
Foundation; he has won more than 30 national and provincial awards
such as the National Science and Technology Progress Second Prize,
and the relevant results have been reported by BBC, The Times. And
CCTV\'s "Science and Technology Expo", "Into the Science" column,
"Guangming Daily" and "Shaanxi Daily" and many other news media
reports. He has undertaken a number of national and provincial
teaching and research projects; won 3 second prizes for national
teaching achievements; published more than 30 textbooks, including 5
national planning textbooks; 2 national quality resource sharing
courses, 1 national bilingual course There are 7 provincial-level
boutique resource sharing courses. '

tokens = nltk.word_tokenize( para3 )
```

4. Porter Stemmer

The Porter English stemmer was described in lectures. There is an implementation of it in NLTK:

```
from nltk.stem import *
stemmer = PorterStemmer()
stemmer.stem( 'dogs' )
```

Try writing code to stem para3 above. Is the stemming correct? Do you see any mistakes?

5. Frequency Distribution

Write a function called freq_dist() to read in a file of English text, tokenise with NLTK, bring down to lower case (Porter will do that for you), stem with Porter and then compute a frequency distribution of the stems. Write out the frequency distribution.

e.g. suppose the input is `'The dogs were watching the cats. One dog watched one cat very closely.'` The output would be:

```
.          2
cat        2
close      1
dog        2
one        2
the        2
veri       1
watch      2
were       1
```

You can download nwu_text.txt from Moodle.

freq_dist() should be called with one parameter which is a filename:

```
>>> freq_dist( 'nwu_text.txt' )
```

It should write to the standard output, i.e. using print() statements in the program.

When you have the program running, save the output and add it as a comment to the end of your program:

```
'''
.          2
cat        2
close      1
dog        2
one        2
the        2
veri       1
watch      2
were       1
'''
```

Hint: You can conveniently use a dictionary to store the frequencies of words:

```
freq = {}
```

```
freq['cat' ] = 0
freq['cat' ] += 1
```

This method allows a default value in the case where it is not in the dictionary:

```
freq.get( 'dog', -1 )
```

You can sort by keynames like this:

```
for x in sorted( freq ):
    print( x )
```

Note this does not sort the dictionary itself. Dictionaries are unordered.

You can do your frequency distribution in other ways if you prefer.

6. Upload your program to Moodle

Call your program freq_dist.py

On Moodle, go to Week 2, i.e. 6 September - 12 September. Click Week 2 Lab 1. Upload your file freq_dist.py there.

Please check you have:

- Comment at the start (marks for this) with your name and number
- Code for the function freq_dist()
- Output from freq_dist('nwu_text.txt') added as a comment at the end of the program