Information Retrieval Lab 2

1. Objectives

- Study Chinese dataset
- Read a file, convert to text, tokenise with Jieba
- Start to build an inverted index

2. Study a Chinese Dataset

From Moodle, download taobao_chinese_collection.zip and study it carefully.

.zip contains a project_docs directory with many files in it. Unpack this in 'ir' directory in 'lab' directory so that an .htm file has a path like this:

c:\lab\ir\project_docs\315消费电子投诉网_2017115146.htm

1. There are 20 .txt files. Open one and look at it.

First line is a query, like 淘宝的诞生.
Each remaining line is an .htm document (filename) which answers that query.

2. There are many .htm files, like 马[淘宝商城租金单位]_2017115146.htm
Each may contain the answer to a query.

3. There are many folders (directories) like马[淘宝商城租金单位]_2017115146_files
Each contains files which are needed to make the .htm page display correctly. We do not use these folders in this lab.

3. Read a File, Convert to Text, Tokenise

We want to open an .htm file, parse the HTML, extract the text and tokenise with Jieba.

Download tcf_tokenise_chinese_file.py

This program will do some useful things:

tcf_parse_html_string(doc_html_text)

Parse a string of text which contains an HTML document.
Extract text and discard HTML

class MyHTMLParser(HTMLParser)

Parser, tailored (sort of) to Baike files.

Try the parser in Python. It needs a string containing HTML:

```
import tcf_tokenise_chinese_file
tcf_parse_html_string( '<html><head><title>Test</title></head><body><p>Parse
me!</p></body></html>' )
```

[It only extracts text in ... tags.]

e.g. try this to see:

tcf_parse_html_string('<html><head><title>Test</title></head><body><h1>Parse me!</h1></body></html>')

Now, we have a file and we want to open it, parse HTML and tokenise.
File must be in project_docs directory below our current directory:

tcf_read_file('315消费电子投诉网_2017115146.htm')

Finally, we can use glob to process many files according to their path:

tcf_glob_files()

This uses a path: 'project_docs*.htm' so it will process all the .htm files in the project_docs directory.

As written, it simply writes the filename. You can change it to do what you want.

4. Build an Inverted Index

First, we want to:

- Go through all the .htm files in project_files
- Parse each file, extract the Chinese words using Jieba
- Put all the words together, for whole document collection
- For whole document collection, compute the frequency of each word

- this is the Document Frequency **DF** (simpest form) for each word

1/DF is the Inverted Document Frequency **IDF**

We can create two simple dictionaries for DF and IDF:

df = { 'cat': 7, 'dog': 14 }

idf = { 'cat': 0.143, 'dog': 0.071 }

[Each word in the document collection has one entry in the idf dictionary.]

Then we want to:

- Go through all the .htm files in project_files
- Parse each file, extract the Chinese words using Jieba
- In a file, compute the frequency of each word

This is the Term Frequency **TF** (simplest form) for each word in a document:

```
index = { '315消费电子投诉网_2017115146.htm': { 'cat': 4, 'dog': 6 },\\
          '7天无理由退换货_2017115146.htm': { 'cat': 3, 'dog': 8 }, ....}
```

[Each .htm file in project_docs has one entry in the index dictionary. Each entry is itself a dictionary giving term frequencies for each word in that .html file.]

Finally, we want to invert the index:

```
inverted_index = { 'cat': { '315消费电子投诉网_2017115146.htm': 4,\\
                            '7天无理由退换货_2017115146.htm': 3 },
                   'dog': { '315消费电子投诉网_2017115146.htm': 6,\\
                            '7天无理由退换货_2017115146.htm': 8 } .... }
```

5. Files and Functions

Call your program create_index.py

create_index.py should contain:

df - global variable for document frequency
idf - global for inverted document frequency
forward_index - global for forward index
inverted_index - global for inverted index

The following functions can take any parameters you wish and can communicate using the above globals:

create_df_dictionary() - returns the document frequency dictionary as above.

create_idf_dictionary() - returns the inverted document frequency dictionary

create_forward_index()

create_inverted_index()

Alter tcf_tokenise_chinese_file.py as necessary and upload it as well.

To make index.txt, open the file and use the normal Python print function to print df, idf, forward_index, inverted_index.

6. Upload your program to Moodle

Call your program file create_index.py
Call your output file index.txt

On Moodle, go to Week 3, i.e. 13 September - 19 September. Click Week 3 Lab 2. Upload your files create_index.py, tcf_tokenise_chinese_file.py and index.txt there.

Please check you have:

- Comment at the start (marks for this) with your name and number
- Code for the functions specified (use other functions / classes in addition as you wish)
- Global variables as specified
- Output in index.txt as specified