CAB431 Tutorial (Week 4): Pre-Processing: Stemming

****

PRELIM 1. Poter2 Stemmer

Stemming refers to a crude heuristic process that removes the ends of words in the hope of finding the stemmed common base form correctly most of the time. This process often includes the removal of derivational affixes. Lovins (1968) defines a stemming algorithm as "a procedure to reduce all words with the same stem to a common form, usually by stripping each word of its derivational and inflectional suffixes (and sometimes prefixes)". A common stemming algorithm is the Porter2 (Snowball) algorithm. You can read the details of the Porter2 algorithms form here:
http://snowball.tartarus.org/algorithms/english/stemmer.html

Download and import Snowball Stemmer into your Java project.

The Snowball Stemmer for java is available here:
http://snowball.tartarus.org/download.php
http://snowball.tartarus.org/dist/libstemmer_java.tgz

Steps to import Snowball Stemmer into your java project (NetBeans)

- Unzip the downloaded libstemmer_java file, find the folder of ***org*** inside the *libstemmer_java\libstemmer_java\java*.
- Copy org folder and paste into your ***src*** folder of your *cab431tut* project
- Start/restart NetBeans IDE, fix the errors such as removing unreachable statement, compile your project until no error message displayed.
- Read TestApp.java source code, learn how to use Snowball Stemmer in your java class.

```
Class stemClass = Class
        .forName("org.tartarus.snowball.ext.englishStemmer");
SnowballStemmer stemmer = (SnowballStemmer) stemClass.newInstance();
stemmer.setCurrent(word.toLowerCase());
```

```
stemmer.stem();
stemmer.getCurrent();
```

If you are using Python, please go to https://pypi.python.org/pypi/stemming/1.0 to download Python implementations of porter2 stemming algorithms, follow the instruction to import and use stemmer in your Python code.

If you use C#, you can find a simple ready-to-use c# class for port2 stemming at:

http://snowball.tartarus.org/otherlangs/english_cpp.txt

You could just copy and paste the source code text to a c# class in your c# project.

TASK 1: Stemming – using porter2 stemming algorithm to update BowDocument's term list (e.g., HashMap or dictionary)

- Create a method in your processor class:

```
/**
 *
 * @param stemmer a SnowballStemmer
 * @param word a String
 * @return a stemmed form of given word
 */
private String stemWordBySnowball(SnowballStemmer stemmer, String word) {
        //your code here
}
```

- Declare and initialize a SnowballStemmer variable and call above method to stem every term before storing it in BowDocument. (if you have not remove less-than-three-letters-word in last week tasks, then you can remove them in this week tasks)
- Output updated term list of every BowDocument in the collection of 10 given input files.
- Compare with last week output to see what a stemmer doing in text pre-processing.

How to submit:

Please save your output into a text file (file name is your full name_wk4, e.g., Yuefeng Li_wk4.txt) and zip codes into another file (e.g., Yuefeng Li_wk4.zip), then send both of them to y2.li@qut.edu.au

Examples of output

```
Document 809495 contains 243 terms and have total 703 words.
israel:13
quot:13
lebanon:11
isra:9
attack:8
civilian:7
hizbollah:7
rocket:5
palestinian:5
netanyahu:5
katyusha:4
side:4
jezzin:4
armi:4
pro:4
violenc:3
kill:3
northern:3
bomb:3
quiet:3
militia:3
presid:3
author:3
tuesday:3
shell:3
town:3
```

arafat:3

respons:3

wound:3

monday:3

secur:3

lebanes:2

offici:2

month:2

islam:2

shoot:2

minist:2