LAB 1

Based on lecture 3 and 4

PREPROCESSING

- You need to have Perl or Python on your machine (you still can use something else).
- Download the following files:
  - Collection 1 --> Bible: link
  - Collection 2 --> Quran English translation: link
  - Collection 3 --> Wikipedia abstracts: link
  (note: if the downloaded abstracts.wiki.txt.gz file size is over 470MB, then it should be uncompressed already. Just rename it to .txt file without the need to uncompress it).
- Write code to do the following:
  - **Tokenisation**: convert text into tokens with no puncituations
  - **Case folding**: make all text into lower case
  - **Stopping**: remove English stop words
  - **Normalisation**: Porter stemmer at least. You can try other stemmers as well. You can get Porter stemmer in Perl or Python, or you can use Snowball Stemmer for multiple languages.
- Print new files for collection 1, 2 and 3 after preprocessing.
- Discuss with your colleagues, what kind of modifications in preprocessing could be applied. For example:
  - Additional words/terms to be filtered outline
  - Special tokenisation
  - Additional normalisation to some terms

TEXT LAWS

For each of the three collections,

- Print the unique terms with frequency, then plot them in a log-log graph. Report with your friends what you notice on Zipf's law
- Plot the distribution of the first digit in frequences obtained and observe Beford's law. Try again while neglecting the one digit frequencies (frequencies less than 10), and check if the law still applies.
- Plot the growth of vocabulary while you go through the collection and observe Heap's law. Try to fit the law to you graph and report the best fitting *k* and *b* constants.
  Advice on how to implement:
  - read text file term by term. count *n* (the number of terms read).
  - save new terms in a hash as you go in reading the file. With each new term update the vocabulray size *v*.
  - print the values of *n* and *v* every while. Plot *n* vs *v* at the end.
  - try to fit an equation $v = k.n^b$. Report best fitting *k* and *b*.

USEFUL TIPS

Print frequency of unique terms in a given collection:
- cat text.file | tr " " "\n" | tr "A-Z" "a-z" | sort | uniq -c | sort -n > terms.freq
- cat text.file | perl -p -e "s/[^\w]+/\n/g" | tr "A-Z" "a-z" | sort | uniq -c | sort -n > terms.freq

All Unix Shell Commands for Windows:
- download: here
- unzip the directory at a decent location on your drive (e.g. c:\ or c:\program files\)
- add the path to the "bin" directory to your Windows path: (example)

Informatics Forum, 10 Crichton Street, Edinburgh, EH8 9AB, Scotland, UK
Tel: +44 131 651 5661, Fax: +44 131 651 1426, E-mail: school-office@inf.ed.ac.uk
Please contact our webadmin with any comments or corrections. **Logging and Cookies**
Unless explicitly stated otherwise, all material is copyright © The University of Edinburgh