**Source**

**Retrieving Data**

The brief asked to compare three different topics (two of similar fields and one unrelated field). In order to have a good starting point for the preprocessing phase I decided to choose topics I had a particular interest in with in depth knowledge of each field. Designing a model to predict topics would be difficult If I had little or no understanding of words or phrases that may help identify each individual topic.

I chose two computer science related topics in GNU and UNIX. I sourced text for the GNU documents from [www.gnu.org](http://www.gnu.org). Similarly I sourced text for the UNIX documents from [www.unix.org](http://www.unix.org). For the third topic I chose match reports for Liverpool Football Club. A topic disconnected from computer science but something I have a lot of knowledge about. I sourced these reports from [www.premierleague.com](http://www.premierleague.com).

For the UNIX and Liverpool FC topics I manually copied and pasted the text in to individual .txt files with names from 0-12. For the GNU topic I used a rapid miner web crawl process. All of the text documents were verbose averaging over 500 words per document.

**Web Crawler**

To retrieve the GNU text files I implemented a web crawler within rapid miner to process documents form the web. I set the URL to [www.gnu.org](http://www.gnu.org) and created regular expression crawling rules to prevent the crawler from retrieving undesirable text.

A feature of the GNU website which allowed a user to select from 16 different languages through hyperlinks resided at the top of each gnu.org web page. This posed a problem as I was using the parameter “store\_with\_matching\_url”, the only difference with each URL was that two letters representing the displayed language were incorporated in to each unique URL.

To over come this problem I ended up setting the max pages parameter on the process to 29 (16 languages + 13 web pages). This allowed my “store\_with\_matching\_url” parameter to overcome language select option.

I used a regular expression of “http://www.gnu.org/gnu/.+”, This rule ensured that the web crawler did not store any external links but only web pages within the GNU domain. Some of the links on the webpage were administration links to gnu.org servers, these pages also contained undesirable content.

**Organising Text Files**

It is desirable to have as many documents as possible to truly develop a model that can analyse text and distinguish unique topics. This brief asked for 13 documents to be sourced, separating 10 files for training and 3 files for testing.

For training purposes I needed to create 3 individual folders containing text documents for each topic. As I needed to test the model on similar data I randomly removed 3 documents from each folder and combined them in to a new unlabelled folder.

At this stage I had 4 folders, 3 folders containing 10 documents with known topics and 1 folder containing 9 documents with unknown topics.

**Word Cloud**

To create word clouds for each topic I used an online word cloud generator at [www.wordclouds.com](http://www.wordclouds.com). I simply copied and pasted the text from all the files on each topic in to an input box and saved the output .png files.

The word cloud provides a simple visual to help distinguish the occurrence of terms in text. The larger the word appears in the cloud, the more frequent that word is in the input text. Along with knowing the topics, the word cloud sets a good starting point to begin developing a model to analyse documents looking for most frequent words.

****

Figure 1 GNU

The most frequent word in the GNU text files is “free/Freedom”. This is to be expected as the GNU OS stems from the free software movement. The problem here is that free is a very common word outside of the GNU topic, this may not be a helpful word in the testing model.

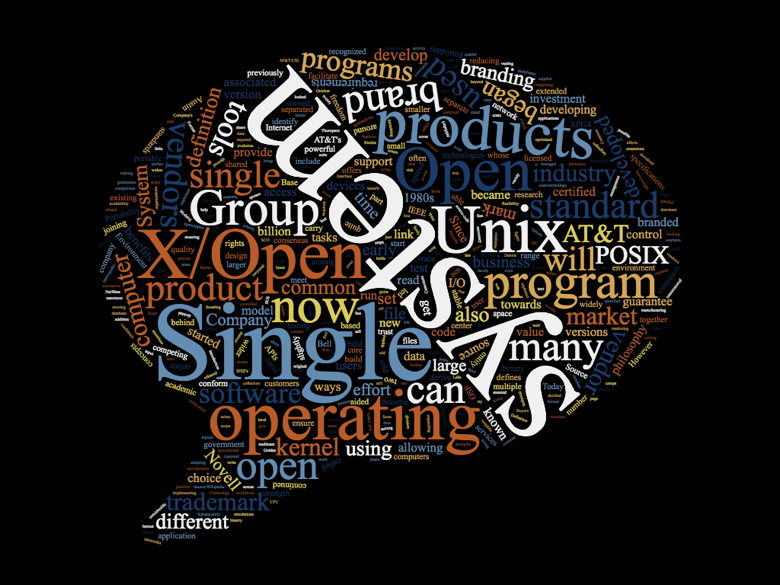


Figure 2 UNIX

The most frequent word in the UNIX text files is “system”. As UNIX is an operating system this again makes sense. Among many disconnected topics, this may help distinguish UNIX text files but in this model the UNIX text files are being compared with GNU text files. So perhaps “system” wont be as helpful at distinguishing UNIX texts as it may seem.



Figure 3 Liverpool

In the Liverpool FC text files, unsurprisingly the most frequent term is “Liverpool”. As this topic is the most disconnected of the three topics, perhaps “Liverpool” can be a helpful distinguishing term. Other common terms include player names.

**Model**

**Bag of Words**

Extract Content

Filters

Phrases

Stemmers

Stem Dictionary

Stop Words

**Document Vector**

Term Counts

TF-IDF

Term Occurrences

Binary Occurrences

Pruning

**Building Model**

**Test**

**Applying Model to Unlabelled Documents**