# Report

## Objective

To build two versions of index for a simple statistical retrieval system. In addition each version should have a compressed form and an uncompressed form.

## Description

### Index Version 1

This version should contains the lemma of words in the dictionary and posting list for each term in the dictionary. Posting list should contain the document ids along with the term frequency.

### Index Version 2

This version should contains the stems of words in the dictionary and posting list for each term in the dictionary. Posting list should contain the document ids along with the term frequency. For stemming, is used.

### Compression

To compress the index, gaps in each posting list is calculated. Then gaps are compressed using delta encoding and term frequency is compressed using gamma encoding. After compression index is stored in a separate binary file.

## Implementation

### Index Version 1

The program follows below mentioned steps for creating index from a repository of documents.

1. Parse a document, removes SGML tags from the file using library and convert it into a string.
2. Using library generate lemmas from the string and store into a Map along with frequency of each lemma occurring in that file.
3. Remove stop words from the Map.
4. Create index for this document using the Map.

Above steps are repeated for each file in the repository. This creates the uncompressed version of the index. Write this index to a binary file.

For compression,

1. Calculate gaps in the posting list of each term in the index.
2. Encode gaps using delta encoding and term frequency with gamma encoding.
3. Store this information in a new map.

This creates the compressed version of the index. Write this index to a binary file

## Index Version 2

The program follows below mentioned steps for creating index from a repository of documents.

1. Parse a document, removes SGML tags from the file using library and convert it into a string.
2. Create tokens from this string as done in homework 1 and store them in a map along with the frequency of each token in the document.
3. Apply stemming on each token using and update the map with new stems and new frequencies.
4. Remove stop words from the Map.
5. Create index for this document using the Map.

Above steps are repeated for each file in the repository. This creates the uncompressed version of the index. Write this index to a binary file.

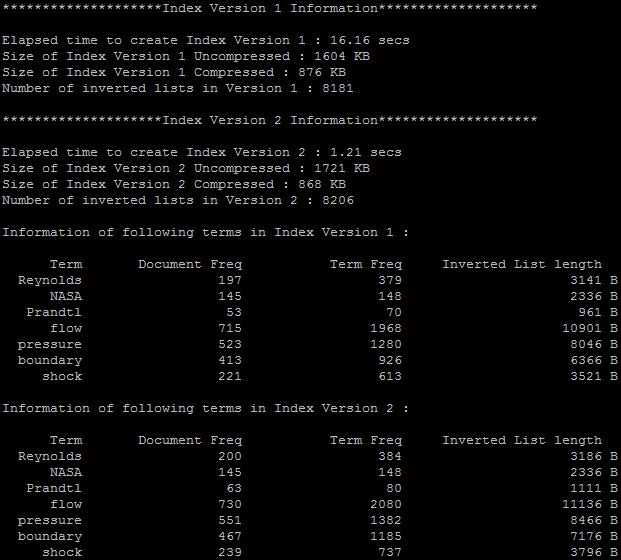
For compression,

1. Calculate gaps in the posting list of each term in the index.
2. Encode gaps using delta encoding and term frequency with gamma encoding.
3. Store this information in a new map.

This creates the compressed version of the index. Write this index to a binary file

## Statistics

Below are the required statistics.



## References

1. <http://chianti.ucsd.edu/svn/csplugins/trunk/soc/layla/WordCloudPlugin/trunk/WordCloud/src/cytoscape/csplugins/wordcloud/Stemmer.java>
2. <http://jsoup.org/>
3. <http://nlp.stanford.edu/software/corenlp.shtml>