Boolean Information Retrieval System

Parth Tulsyan (2020A7PS1883H)
Sreeram Yerasani (2020A7PS0052H)
Prithvi Rajan (2020A7PS2080H

22nd March 2022

1 Overview

This project is aimed to make a stable,simple boolean search system for a fixed corpus of data, i.e 42 William Shakespeare plates, and return the subset of the documents from the corpus satisfying a given boolean condition.

2 Problem Statement

Developing a Boolean Information Retrieval System which returns documents satisfying boolean queries with the following functionalities:

- Stopword removal
- Stemming
- Wildcard Query Handling
- Spelling Correcting: Edit distance (Levenshtein distance method)

3 Architecture

3.1 Corpus Preprocessing

Before starting the querying system we run our code to go through all the corpus and generate the inverted index posting and unique words lists and populate the kgrams list and save the data in json format.

- We used the stopwords ranks.nl. We removed these stop words from corpus.
- We used PorterStemmer from nltk.
- We made a list of unique words for spelling correction while querying.

- We stored the inverted index of documents w.r.t to the stemmed words using a hashtable structure.
- For every unique words we made an inverted index of K-grams in a hashtable data structure which will be used in wildcard querying.

3.2 Query Processing

Before starting the query processing, we take the following steps:

1. Load inverted index of docID in form of linked list.
2. Load unique words list.
3. Load K-grams inverted index in form of linked list.

After inputting the query we convert the query to Postfix format with preference order NOT > AND > OR. Then we traverse the postfix expression and solve it.

1. Check if it is a wildcard or not
2. If it is a wildcard we do wildcard querying using K-grams.
3. If it is not a wildcard we check for spelling mistakes using Levenstein Edit Distance.

3.3 Time Complexity

- Stop word removal - $O(n)$
- Wild card querying - $O(n^2)$
- Spelling correction - $O(n)$
- Merging inverted index - $O(n + m)$

4 Documentation

We used pdocs to make documentation and is stored in /html folder