Birla Institute of Technology and Science-Pilani, Hyderabad Campus

First Semester 2018-2019

# 

# Information Retrieval (CS F469)

# Design Document

# Assignment 1

Domain Specific Information Retrieval System

Rohith Saranga (2017A7PS0034H)

Kasuba Badri Vishal (2017A7PS0270H)

Rikil Gajarla (2017A7PS0202H)

Under the guidance of

Dr. Aruna Malapati

## Abstract:

This project aims to build a domain specific search engine based on vector space model. Initially the documents to be searched are pre-processed to generate tokens. In the vector space model, documents are represented as vectors containing weights of these tokens. When a query is given to the search engine, a vector is made from its tokens and similarity between all the documents is found. Then the documents are returned in order of decreasing similarity.

#### Architecture:

Programming Language: python

Dataset: Journals on Computer Science (350 journals scraped from archive.org)

Some of the python libraries used are nltk, pandas, numpy and pickle.

Folder Structure:

.

├── irstructures

│   ├── \_\_init\_\_.py

│   ├── document.py

│   ├── invertedindex.py

│   └── models

│       ├── \_\_init\_\_.py

│       ├── boolean\_retrieval.py

│       └── vector\_space.py

├── corpus

├── pickle\_files

└── main.py

Classes used are

1. Document
2. Tf\_Idf
3. InvertedIndex

## Working description:

A document object is created by taking text from a file as input. It does preprocessing like tokenization, case normalization, stemming and removal of stop words. It also stores unique words and their frequencies in a dictionary.

All the unique words from all documents are also stored in a dictionary along with their total frequencies.

A document vector is a series of weight values for each term in the corpus. The weighting scheme used is tf-idf (term frequency – inverse document frequency).

##### Tf-Idf Weighting:

|  |  |
| --- | --- |
| Term Frequency =  where t = term t and d = document d |  |
| Document Frequency  Number of documents containing the term t |  |
| Inverse Document frequency  N = number of documents |  |
| Tf-idf weight  for a term, t in document, d | = |

## Similarity Metrics:

A query is converted into a vector with tf-idf weights. A suitable metric like cosine similarity is used to find similarity between this query vector and all the document vectors and they are returned in order of decreasing similarity. The top ten results are returned to the user.

## Efficient approach by augmenting Boolean retrieval model:

Finding similarity of query vector with all other documents in the corpus takes a lot of time. Since boolean retrieval model is very fast compared to vector space model, simililarity can only be calculated for the documents appearing in the output of boolean retrieval model. Now documents can be ranked based on similarity.

Boolean retrieval model requires construction of inverted index which is easy because of the document objects that are already created. This inverted index is used in both places. It is used for the boolean retrieval model and also used to find document frequency efficiently in calculating tf-idf weights.

## Results:

Running times:

* corpus generated in: 77.59436631202698 s
* inverted index generated in: 0.21642398834228516 s
* Data Frame initialized in 2.320979118347168 s
* Data Frame made in 3584.1931648254395 s
* vector space model built in: 3586.5141439437866 s
* Size of dataframe: 93526 terms X 350 docs

Output:

Enter query: UC Berkeley

Tf-Idf results:

corpus\1703.05435.txt    0.06184110510117019

corpus\1704.07123.txt    0.04745490297332865

corpus\1704.06319.txt    0.04729361788006495

corpus\1703.01522.txt    0.035526042336960016

corpus\1701.00251.txt    0.03427717041541585

corpus\1401.6325.txt     0.028963980522957806

corpus\1405.0055.txt     0.028778287694046795

corpus\1703.00986.txt    0.025651801664318963

corpus\1701.05756.txt    0.023710152835947223

corpus\1702.06537.txt    0.022252284010918492

returned in  19.372350692749023 s

performace increased when Boolean retreival model is used

Enter query: UC Berkeley

Boolean Retrieval results:

corpus\1401.6325.txt

corpus\1703.04736.txt

corpus\1701.03297.txt

corpus\1703.05348.txt

corpus\1703.05435.txt

corpus\1701.04525.txt

corpus\1701.05187.txt

corpus\1405.0055.txt

corpus\1703.08661.txt

corpus\1701.05451.txt

corpus\1701.05756.txt

corpus\1703.08823.txt

corpus\1703.10935.txt

corpus\1704.02949.txt

corpus\1702.00320.txt

corpus\1702.00523.txt

corpus\1505.04409.txt

corpus\1704.05560.txt

corpus\1507.03304.txt

corpus\1704.06319.txt

corpus\1704.06611.txt

corpus\1704.07123.txt

corpus\1702.06537.txt

corpus\1512.00482.txt

corpus\1602.04294.txt

corpus\Computer-Aided-Development-of-Fuzzy-Neural-and-Neuro-Fuzzy-Systems-Priti-Srinivas-Sajja.txt

corpus\1703.00092.txt

corpus\1703.00425.txt

corpus\1605.07805.txt

corpus\1703.00986.txt

corpus\1703.01522.txt

corpus\1607.01474.txt

corpus\1701.00251.txt

corpus\1703.03640.txt

34 files returned in 0.0065460205078125 s

Tf-Idf results:

corpus\1703.05435.txt    0.06184110510117019

corpus\1704.07123.txt    0.04745490297332865

corpus\1704.06319.txt    0.04729361788006495

corpus\1703.01522.txt    0.035526042336960016

corpus\1701.00251.txt    0.03427717041541585

corpus\1401.6325.txt     0.028963980522957806

corpus\1405.0055.txt     0.028778287694046795

corpus\1703.00986.txt    0.025651801664318963

corpus\1701.05756.txt    0.023710152835947223

corpus\1702.06537.txt    0.022252284010918492

returned in  2.0605480670928955 s