Information Retreival Project

Rohan Jauhari

March 2021

1 Introduction

Dataset used- The BBC news categorization dataset has been used which has several fields such as Political, Entertainment, Business, Technology and Sports. It can be found out here:http://mlg.ucd.ie/datasets/bbc.html. The dataset is in form of a directory with sub-directories of each field. Each field directory has numerous text files which have news related to that field. These text files are raw. They have punctuation marks, stop words, Capitals etc. It is a semi structured dataset as we have a hierarchy of fields and same kind of documents are placed under same field. Also, the dataset is not completely structured as it doesnt have a proper terminology as to which term is present where and is of what significance.

We have randomly chosen 50 text files from 5 folders. So, a total of 250 text files are here. We will be processing these text files to get as relevant document as possible for user queries.

We will be using python 3.7 on Spyder IDE.

2 Processing

TASK 1 The first step is reading data from the file explorer. We will be reading files using file IO in python and creating an array of text of documents. Then we will be using built in functions for removing stop words, punctuation, tockenization, stemming. We will then get an array of processed text documents ready for further action.

We now create a term document matrix using the built in function and save it as a CSV file whose size is 3536KB. This is created using the de-tockenized tockens as we want strings as a matrix doesnt get formed using tokens via text mining library.

Now, we create an inverted index matrix. We list all the unique terms present in the documents and then for each term, we check it's presence in each document and append the docID if present. Then, we take a dictionary and make keys as terms and append corresponding docIDs as list in values. We then save it as a CSV file which occupies 290 KB.

The CSV Files are present in the attached folder.

TASK 2 Now, to the benchmark queries, are as follows: 1. Tell me about new marriage rules for people coming from foreign to UK. (144) 2. Does DS technology let players take against people wirelessly. And is it on sale in Europe or Japan ? (246) 3. Is it Racism proof to play sports? Are Black players safe? (187) 4. Is there some mouse assistive technology for elderly? (248) 5. Argentina- Venezuela agricultural deal food crisis. (9)

The document that these queries have been fetched from are written in from of them.

We will create relevance judgement for each query document pair. For this, we need to find tf-idf weights for all the documents. We preprocess the query with all steps mentioned above then we check count of total common tokens between each query and document and store it in as a CSV file named 'Relevance Judgement.csv'.

Now, we have to form a ranked retrieval. For this, we form tf-idf weights for both query and all documents and find relevance score based on cosine similarity.

We find term-frequency i.e. count of a term in a document for each query document pair and then find weights by: 1+np.log10(doc.count(term) if term is present in doc or 0 otherwise. Document frequency is number of documents having a term. We also find inverse document frequency by df[i]=np.log10(N/df[i]) for each temm.

Then, in "tfidf matrix" matrix, we multiply tf weights for each term docID pair with df weight for every term. We process following query and document in the same way.

We take query Query- Does DS technology let players take against people wirelessly. And is it on sale in Europe or Japan ? (246)

to find ranked retrieval of documents. We will now find the cosine scores for each document with the query.

For each token in query, we find the postings list of it. We get DocIDs of them. and do the following: cosine$_{s}cores[docId]+ = tfidf - query[term] * tfidf - doc[(term, docId)]$

We will be iterating all postings list and will keep updating the cosine score by adding the product of tf-idf scores of query term and term DocID pair. We will then divide each document's score by length of that document's tokens to length normalize as follows. cosine-scores[i+1]/=len(stemmed-array[i])

We then get cosine scores of all documents and get a ranked retrieval. Higher the score, higher is the relevance.

Now the documents are 246, 28, 222, 212, 233, 56, 146, 24, 217, 227. The last 50 documents i.e. 200-250 are of tech. The query is of tech. It talks about tech sales in countries.

Calculating Average Precision:- Precision=fraction of retrieved docs that are relevant. The documents are R,NR, R,R,R,NR,NR,NR,R,R. Total 6 relevant documents.

so, the precision for respective documents in the above ranked retrieval is 1,1/2,2/3,3/4,4/5,4/6,4/7,4/8,5/9,6/10 so, the average precision is (1+2/3+3/4+4/5+5/9+6/10)/6= 0.728

The document which the

Calculating mean average precision:- we will be taking five queries

1. Tell me about new marriage rules for people coming from foreign to UK. (144)

Relevant documents are 144,150, 21, 55, 134, 71, 56, 248, 216, 159 The documents are R,R,NR,NR, R,NR,NR,NR,NR so, the precision for respective documents in the above ranked retrieval is 1,2/2,2/3,2/4,3/5,3/6,3/7,3/8,3/9 so, the average precision is (1+2/2+3/5)/3=0.867

3. Is it Racism proof to play sports? Are Black players safe? (187) Relevant documents are 222,110,196,187,158,186,191,152,189,206 The documents are NR,NR,R,R,R,R,R,R,R,NR so, the precision for respective documents in the above ranked retrieval is 0,0,1/3,2/4,3/5,4/6,5/7,6/8,7/9,7/10 so, the average precision is (1/3+2/4+3/5+4/6+5/7+6/8+7/9)/7=0.620

4. Is there some mouse assistive technology for elderly? (248) Relevant documents are 248, 238, 220, 214, 230, 221, 216, 137,242,213 The documents are R,R,R,R,R,R,R,NR,R,R so, the precision for respective documents in the above ranked retrieval is 1,2/2,3/3,4/4,5/5,6/6,7/7,7/8,8/9,9/10 so, the average precision is (1+2/2+3/3+4/4+5/5+6/6+7/7+8/9+9/10)/9=0.976

5. Argentina- Venezuela agricultural deal food crisis. (9) Relevant documents are 9,22,159, 21, 6,193, 132, 3,7,25 The documents are R,R,NR,R,R,NR,NR,R,R,R so, the precision for respective documents in the above ranked retrieval is 1,2/2,2/3,3/4,4/5,4/6,4/7,5/8,6/9,7/10 so, the average precision is (1+2/2+3/4+4/5+5/8+6/9+7/10)/7= 0.791

Mean Average Precision=(0.867+0.728+0.620+0.976+0.791)/5=3.982/5=0.7964