# Results

For **query1.txt**, we found that **document1352.txt** had the highest similarity score of 0.0572.

For **query2.txt**, we found that **document1165.txt** had the highest similarity score of 0.0292.

For **query3.txt**, we found that **document348.txt** had the highest similarity score of 0.0512.

The full list of the top 10 most similar documents with each query is in the table below:

|  |  |  |
| --- | --- | --- |
| Query1.txt | Query2.txt | Query3.txt |
| document1352.txt: 0.0572 | document1165.txt: 0.0292 | document348.txt: 0.0512 |
| document1138.txt: 0.0536 | document750.txt: 0.0269 | document2185.txt: 0.0412 |
| document479.txt: 0. 0416 | document913.txt: 0.0262 | document281.txt: 0.033 |
| document1090.txt: 0. 0177 | document2008.txt: 0.0204 | document160.txt: 0.0324 |
| document42.txt: 0. 0153 | document2009.txt: 0.0204 | document161.txt: 0.0323 |
| document247.txt: 0.1633 | document914.txt: 0.0204 | document987.txt: 0.0244 |
| document1342.txt: 0. 0107 | document2184.txt: 0.0178 | document2495.txt: 0.0219 |
| document353.txt: 0. 0089 | document1738.txt: 0.017 | document984.txt: 0.0214 |
| document1211.txt: 0. 0079 | document1907.txt: 0.0169 | document1738.txt: 0.0198 |
| document201.txt: 0. 0079 | document1737.txt: 0.0166 | document1737.txt: 0.0195 |

# Methodology

## Part II: Question 1 and 2

We first created a dataframe containing the Inverse Document Frequency (IDF) of each word in the entire corpus and every document in the entire corpus. We followed the formula outlined in Week 5 slides, slide 28 to calculate IDF.

Table

Description automatically generated

We then processed each word in the corpus and calculated the Term Frequency (TF) for each word in the document and multiplied by the IDF and updated the dataframe to obtain the following which to obtain the following vector space representation of the TF-IDF for each word for each document:

Table

Description automatically generated

In the picture above, ‘authenc’ has a TF-IDF score of 2500 for document1.txt but a TF-IDF score of 0 for document10.txt

We then repeated the afore-mentioned steps for the query files to obtain the following vector space representation for query:

|  |  |
| --- | --- |
| Table  Description automatically generated | Table  Description automatically generated |

## Part II: Question 3

The queries and the documents live in different vector space of different dimensions as the documents and the queries have different vocabularies. As such, we have to construct a new Vector Space that has embedded TF-IDF scores across all documents and queries. As shown below, the TF-IDF scores for all the documents and Queries are calculated together that better represents one coherent vector space.

A black screen with white text

Description automatically generated with low confidence

Combined Documents and Queries dataframe

After we have created new vector space for both the queries and documents combined, we calculate the cosine similarity between the TF-IDF values of every pair of query and document before sorting the scores in decreasing order (as demonstrated with the following code snippet).

Graphical user interface, text, application, email

Description automatically generated