1. Bag of words (BOW) – collate all the distinct words used in all the documents

2. Document term matrix (DTM) –columns are the BOW and rows are your documents. Words/terms are our features of DTM. Values of the matrix is the count of the word in that document

3. Term document matrix (TDM) – it’s the transpose of the DTM

Doc0= "This is a very good and plain paper. This is really good and interesting"

Doc1 = "This paper is very interesting, awesome"

Stop\_Words = ["This","is","a","very","and"]

**Step:1 Remove stop words and punctuations and get a clean doc’s**

doc0\_clean = "good plain paper really good interesting"

doc1\_clean="paper interesting awesome"

Step:2: Obtain the bag of words from the cleaned documents

BOW = {“good”, “plain”, “paper”, “really”, “interesting”, “awesome”}

Step3: compute the document term matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | good | plain | Paper | really | interesting | awesome |
| 0 | 2 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |

4. Term frequency – TF -> occurrence of a word in a doc/total no. of distinct words in that document

TF(good,0) = 2/5

TF(plain,0) = 1/5 = TF(paper,0) = Tf(plain,0)

TF(awesome,0) = 0/5 = 0

TF(paper,1) = 1/3

TF(interesting,1) = 1/3

TF(awesome,1) = 1/3

5. Document frequency = No. of documents containing the word/Total no. of documents

DF(good) = ½

DF(plain) = ½

DF(interesting) = 2/2 = 1

6. Inverse document frequency – IDF = It’s the logarithmic inverse of DF

IDF = ln(1/DF)

IDF(good) = ln(2) = 0.693

IDF(interesting) = ln(1) = 0

7. TFIDF = TF\*IDF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TFIDF = TF\*IDF | |  |  |  |  |  |
|  | good | plain | paper | really | interesting | awesome |
| 0 | 0.277259 | 0.138629 | 0 | 0.138629 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0.231049 |

8. Implementing all these in python