

LAB - 8

Aim: To compute the Term Frequency-Inverse Document Frequency (TF-IDF) values for a book dataset stored locally (book.txt) using Apache Pig.

Procedure:

Step 1: Preprocess the data

1. Store the book dataset (book.txt) in the local filesystem.
2. Ensure that the dataset contains lines of text separated properly for processing.

Step 2: Start Pig in Local Mode

Run Pig in local mode by using the command:

```
pig -x local
```

Step 3: Load the dataset

Load the input file (book.txt) into Pig:

```
docs = LOAD 'book.txt' AS (line: chararray);
```

Step 4: Assign unique document IDs

Generate unique IDs for each document:

```
docs_with_id = RANK docs;
```

Step 5: Tokenize the text into words

Split the lines into individual words:

```
docs_split = FOREACH docs_with_id GENERATE RANK AS doc_id,  
FLATTEN(TOKENIZE(line)) AS word;
```

Step 6: Calculate Term Frequency (TF)

1. Group by doc_id and word:

```
grouped = GROUP docs_split BY (doc_id, word);
```

2. Count occurrences of each word within the document:

```
word_count = FOREACH grouped GENERATE FLATTEN(group) AS (doc_id, word),  
COUNT(docs_split) AS term_frequency;
```

Step 7: Calculate Document Frequency (DF)

1. Remove duplicate word-document combinations:

```
unique_words = DISTINCT docs_split;
```

2. Group by word and count the number of documents containing each word

```
doc_freq = FOREACH (GROUP unique_words BY word) GENERATE group AS word,  
COUNT(unique_words) AS document_frequency;
```

Step 8: Calculate Total Number of Documents (N)

```
total_docs = FOREACH (GROUP docs_with_id ALL) GENERATE COUNT(docs_with_id) AS  
total_documents;
```

Step 9: Compute TF-IDF

```
tfidf = FOREACH joined GENERATE word_count::doc_id, word_count::word,  
word_count::term_frequency * LOG(total_docs.total_documents /  
doc_freq::document_frequency) AS tfidf_value;
```

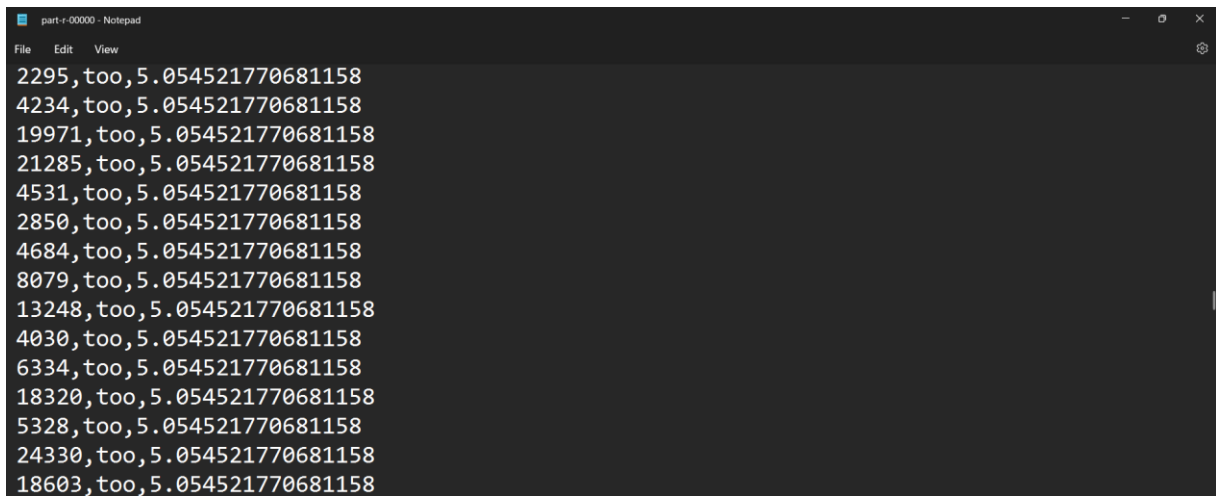
Step 10: Store the Output

Store the results in the local filesystem:

```
STORE tfidf INTO tfidf_results' USING PigStorage(',');
```

Output Screenshots

Tfidf File



The screenshot shows a Notepad window titled 'part-r-00000 - Notepad'. The text inside the window lists 18 lines of TF-IDF values for the word 'too'. Each line consists of a document ID, the word 'too', and a numerical value. The values are all identical, representing the inverse document frequency (IDF) of the word.

```
2295,too,5.054521770681158
4234,too,5.054521770681158
19971,too,5.054521770681158
21285,too,5.054521770681158
4531,too,5.054521770681158
2850,too,5.054521770681158
4684,too,5.054521770681158
8079,too,5.054521770681158
13248,too,5.054521770681158
4030,too,5.054521770681158
6334,too,5.054521770681158
18320,too,5.054521770681158
5328,too,5.054521770681158
24330,too,5.054521770681158
18603,too,5.054521770681158
```

Results and Conclusion

- The TF-IDF values for each word in the documents were successfully computed.
- High TF-IDF values indicate words that are important to specific documents but not common across the book.