**Contents**

1. Introduction and Problem Statement
2. Dataset Details
3. Preprocessing Dataset
4. Model Creation
5. Results
6. References

**1. Introduction**

Source data from Carnegie Movie Summary Corpus site. Where movie plot summaries are considered.

<http://www.cs.cmu.edu/~ark/personas/>

**Problem Statement**:

The objective of this project is creating a search Engine. We have considered the movie plot summary dataset from Carnegie Movie Summary. This dataset has ID for a movie and the description for that movie. From this we can do text classification and by performing different techniques We create a search engine. By querying which are either single or multiple terms we can extract the similar documents.

**2. Dataset Details**

The dataset is a movie plot summary which are extracted from Wikipedia, aligned metadata has been extracted from freebase.

The dataset consists of movie box office revenue, genre, runtime and language. Along which it has the character names and information about the actors that tells their age and gender when movie was released. Hence, we have extracted the entire data from plotsummary.txt file.

**3. Pre-Processing**

**Text Mining in R:**

We removed stop words using tm package. Corpus is the structure of managing the data, which is the collection of text document.

Several thousand documents in the dataset were considered and its description was viewed using inspect.

For preprocessing we have done stemming using corpus package.

Removing stop words using tm\_map , where whitespaces stem documents were removed.

Term document was created using the rows and by normalizing the data by applying td idf weighing matrix. This generated term document has been transformed into a dataframe.

We have used DTM ( Document term Matrix) where each row represents a matrix, column represents one term and each value contains the appearance that occurs. We have used tidy() which is document-term matrix into a tidy data frame. Similarly the cast() that has one term per row.

**4.Model Creation**

**Tf-idf technique:**

 Tf-idf with full form as Term frequency–inverse document frequency helps in predicting the importance of a word in a document or a corpus.

The information retrieval is done by using weighing factors. TF-idf is proportional to the number of similar words in the document. Validating this tf-idf helps in the search engine, where we can find the frequency of a word. We have used this in stop word filtering and summarizing the text. So this uses term frequency and inverse term frequency to identify the word.

By this we can get. The frequent word in the document which is in English like “the” “is” So these kinds of words have been added to the stop list. Accordingly we have adjusted the term frequencies to the most frequently used item.Similarly, the inverse document frequency is where the weight of the commonly used word is decreased, and that word wouldn’t be considered.

We have created a most frequently used words cloud using word cloud in R. Where the top frequently used words in the document has been seen in the below picture.

**The Most frequently occurring words in the document (word cloud ) :**



From this we can notice that kid the and play etc. occur more frequently in the plotsummary.txt file.

We are using the sparse matrix where the data is dynamically loaded into it. As it is the sparse matrix the data is very efficient.

We have calculated top ten results by cosine similarity. By calculating the similarity matrix between all column vectors of a given matrix. This measures the similarity between two vectors of the product and the angle between them. As this can be applied to any number of dimensions except that they should be non-zero vectors.

**5. Results:**

We have read different queries and have retuned the output and the cosine scores of it.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Query** | **Result** | **Score** |
| 1 | Funny Movie | 1.Paschalis is a 30 year old agronomist about  2. After 29 years of marriage Alberto  3. Returning to her native Turin for the opening  4.Odile Deray is the publicist for the slasher  5.The movie is set in a small rural town  6.Stratis after many years of laborious work  7. Jackie York is a famous  8. The central character is the Post Master  9. It is New Years Day of 2006  10. The film is based rather loosely on the play | 1 0.1364  2 0.0986  3 0.0780  4 0.0537  5 0.0319  6 0.0242  7 0.0202  8 0.0190  9 0.0182  10 0.0177 |
| 2 | zorro legend | 1.finished filling little Marianne Edwards favorite  2. Unlike the original stage version the main plot of the film  3. Danny Stetson  is a clean cut allAmerican college student | 1. 0.0305  2. 0.0196  3. 0.0132 |
| 3 | Games | 1. The five richest men in the  2. A group of criminal aliens called the The Nerdlucks led by their boss Mister Swackhammer  3. An unidentified flying object makes an emergency landing on Earth and is taken into custody by the United States government  4. Hiroki Imanishi and Kanata Tokino are two young  5. Nobita and Doraemon travel to the future to see whether Nobita will marry  6. Alex Kovac playing poker in New York City drops  7. Thomas Crown is a wealthy financier who aches for  8. Wally had finished filling little Marianne Edwards  9. Shankar a petty thief in desperate need of money steals  10. Beth Cappadora and her husband Pat experience | 1 0.0924  2 0.0740  3 0.0720  4 0.0665  5 0.0569  6 0.0558  7 0.0551  8 0.0541  9 0.0517  10 0.0494 |
| 4 | Friend | 1. A womanising playboy becomes tired  2. Newly commissioned Guards Ensign  3. he story is about two youngsters a boy  4. Jerry Logan is a Las Vegas  5. The documentary is the official film biography of  6. The film is set in present day Beirut in the midst  7. Randy Mason is a teenage tech whiz who  8. Roberta Peter and Phyllis live a comfortable  9. The film explores the lives of the | 1 0.0718  2 0.0697  3 0.0584  4 0.0518  5 0.0517  6 0.0495  7 0.0444  8 0.0407  9 0.0387  10 0.0380 |
| 5 | Police | 1. Jerry Logan is a Las Vegas police officer  2. In 1942 a 10 year old boy named Timmy plays with a jigsaw puzzle  3. Jane Alexander is a sheltered attractive  4. In the fictional town  5. Pierre suffers from amnesia after a wartime accident in which  6. While the novel is told by a thirdperson  7. Raj Kumar is an honest handsome and intelligent  8. The film opens with a shot of a naked woman lying dead  9. Plot Gigolos in Europe are being terrorized by a  10. Burton Grant exposes politician Jarvis McVey as a | 1 0.1454  2 0.0629  3 0.0610  4 0.0531  5 0.0505  6 0.0453  7 0.0436  8 0.0430  9 0.0413  10 0.0412 |
| 6 | Police games | 1. The five richest men in the territory  2. Jerry Logan is a Las Vegas police  3. A group of criminal aliens called the The Nerdlucks  4. An unidentified flying object makes an emergency landing  5. Thomas Crown is a wealthy financier who  6. Hiroki Imanishi and Kanata Tokino are  7. Nobita and Doraemon travel to the future  8. Shankar a petty thief in desperate need of money steals  9. Alex Kovac playing poker in New York City  10. A Danish officer Michael is sent to the International | 1 0.0808  2 0.0705  3 0.0647  4 0.0630  5 0.0630  6 0.0582  7 0.0556  8 0.0506  9 0.0488  10 0.0481 |

Here, The results of different queries (Both multiple terms and single terms) are given.

When the user gives the values in command line say “Funny movies” It will calculate the cosine similarity with the documents in the file. Depending on the cosine similarity value the results will be displayed.

In the table we have shown the starting of each document as each of the document is very large.

Hence, We have found the best results for the queries.

**6. References:**

<https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf>

<https://www.rdocumentation.org/packages/lsa/versions/0.73.1/topics/cosine>

<https://www.tidytextmining.com/tfidf.html>

<https://www.r-bloggers.com/build-a-search-engine-in-20-minutes-or-less/>

<https://rpubs.com/ftoresh/search-engine-Corpus>

<https://www.datacamp.com/community/tutorials/sem-data-science>

<https://cran.r-project.org/web/packages/stopwords/stopwords.pdf>