Design a Text Retrieval System for Movie Recommendations

TECH3300: Machine Learning Application

Kaplan Business School

2024

Introduction

Objective:The primary objective of this project is to design a text retrieval system capable of recommending movies or shows based on their descriptions. By using text processing techniques and the TF-IDF (Term Frequency-Inverse Document Frequency) algorithm, we aim to build a system that efficiently matches movies based on the semantic content of their descriptions.

Dataset Description:

- **Source**: The dataset is sourced from Kaggle IMDb Movies/Shows.
- **Files Provided**: Two files are provided: train.csv and test.csv. They both have the title and the description of the movie/Show.

Data Loading and Preprocessing

Loading Data:

- Use pandas library to load both the train and the test datasets provided.

Preprocessing Steps:

- Remove NaN and Empty Values: Rows with NaN and empty values are dropped in the title and description columns.
- Text Normalization: Converted text to lowercase and Removed punctuation.
- Tokenization and Stopwords Removal: Split text into words or tokens and remove common words that do not contribute to the meaning (e.g."the", "and")

TF-IDF and Inverted Index

- **TF-IDF Overview:** TF-IDF stands for Term Frequency-Inverse Document Frequency. It's a numerical statistic that reflects the importance of a word in a document relative to a collection of documents (corpus) **Chiny et al.**; **2022**.

- **Term Frequency (TF):** Measures how frequently a term appears in a document. It is calculated as:

$$\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

- **Inverse Document Frequency (IDF):**
Measures how important a term is across the entire corpus. It is calculated as:

$$\text{IDF}(t) = \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing term } t}\right)$$

- **TF-IDF Score:** Combines TF and IDF to give the importance of a term in a document relative to the corpus:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

- **Inverted File:** A data structure used to map terms to the documents that contain them, enabling efficient retrieval.

Finding Similar Movies/Shows

Cosine Similarity:
Used to measure the similarity between two text descriptions **Singh et al.**; **2020**.

Steps:

1. Compute TF-IDF vectors for each description.
2. Calculate the cosine similarity between the test description and each train description.
3. Retrieve the top 3 matches based on similarity scores.

Performance and Results

Performance Evaluation Metric:

- **Average Cosine Similarity:** Measures the average cosine of the angle between the TF-IDF vectors of the test movie descriptions and their top 3 most similar training movie descriptions.
- Average TF-IDF Cosine Similarity: 0.18
- The system achieved a reasonable average cosine similarity score of 0.18, indicating its effectiveness in identifying similar movies based on descriptions.
- Potential improvements include exploring more advanced models such as Word2Vec and integrating additional metadata like genres, cast, and ratings.

Example 1 Result:
Test Movie: The Trip

- **Description**: A collage of film images and ambient dance sounds from Jacques Peretti and DJ Downfall.
- **Top 3 Similar Movies**:
  1. **Mesrine: Killer Instinct**: Jacques Mesrine becomes France's most-wanted criminal.
  2. **Oceans**: Most of the Earth's surface is covered by water; using the latest technology, filmmakers Jacques Perrin and Jacques Cluzaud set out to explore the underwater world. Diving deep into the waters that ultimately sustain all life on Earth, Perrin and Cluzaud capture spectacular footage of the amazing beauty and harsh reality of life beneath the waves.
  3. **The Names of Love**: A sheltered scientist (Jacques Gamblin) and a sexy political activist (Sara Forestier) have a May-December romance.

Example 2 Result:
Test Movie: Bommarillu

- **Description**: Siddhu's overprotective father decides to get him married to a rich girl. Siddhu's life, however, takes an interesting turn when he meets and falls in love with Hasini.
- **Top 3 Similar Movies**:
  1. **Thammudu**: The youngest son of the family does not take his life seriously; his father always reprimands him but his elder brother and a friend love him. He falls in love with a girl and lies to her that he belongs to a rich family.
  2. **Socha Na Tha**: A boy is forced meet a girl and get married to her but even though he refuses to do so, they become good friends. However when the both of them get engaged to different people they realize that they've fallen in love with each other.
  3. **The Good Guys**: Two longtime friends search for a way to get rich.

Conclusion

Summary:

- Built a text retrieval system using TF-IDF and cosine similarity.
- Successfully identified similar movies based on descriptions.

Future Work:

- Explore advanced techniques like Word2Vec.
- Integrate additional metadata such as genres, cast, and ratings .

References

[1] Park, K., Hong, J.S., & Kim, W. (2020). A methodology combining cosine similarity with classifier for text classification. *Applied Artificial Intelligence*, 34(5), pp. 396-411.

[2] Chiny, M., Chihab, M., Bencharef, O., & Chihab, Y. (2022). Netflix recommendation system based on TF-IDF and cosine similarity algorithms. *no. Bml*, pp. 15–20.

[3] Singh, R.H., Maurya, S., Tripathi, T., Narula, T., & Srivastav, G. (2020). Movie recommendation system using cosine similarity and KNN. *International Journal of Engineering and Advanced Technology*, 9(5), pp. 556-559.

[4] Sarica, S., & Luo, J. (2021). Stopwords in technical language processing. *Plos One*, 16(8), p. e0254937.