Build Semantic Search in Moives using NLP with LSI Model and Pretrained LaBSE

Introduction

This project involves developing a semantic search pipeline to extract relevant keywords from English articles and evaluate different models for text similarity. The main objective is to compare traditional NLP methods with deep learning models to determine which provides the best performance in retrieving relevant information.

Data Description

The dataset used for this project is from the Kaggle dataset Movies Similarity, which includes:

rank: The rank of the movie based on ratings or popularity.

title: The title of the movie.

genre: The genre(s) of the movie (e.g., Action, Comedy, Drama).

wiki_plot: The plot summary of the movie from Wikipedia.

imdb_plot: The plot summary of the movie from IMDb.

For this project, we primarily focused on the wiki_plot and imdb_plot features, as they contain the textual data necessary for semantic search.

Baseline Experiments

Goal

The goal was to establish a baseline for semantic search using traditional text processing methods.

Experiments

1. **Data Preprocessing:**

Tokenization, cleaning, removal of stopwords and punctuations using **spaCy**.

2. **Feature Extraction:**

Building a **Bag-of-Words** (BoW) model.

Creating a dictionary and filtering out rare and common terms.

3. **Model:**

Term Frequency-Inverse Document Frequency **(TF-IDF) model.**

Latent Semantic Indexing **(LSI) model** for dimensionality reduction.

Conclusions

The baseline models provided initial results on the effectiveness of traditional text processing methods for semantic search. The TF-IDF and LSI models were used to retrieve relevant documents based on input queries.

Other Experiments

Experiment 1: Optimizing Search with LSI Model

Goal: Improve search results by leveraging the LSI model's ability to capture latent semantic structures.

Steps:

1. Built the TF-IDF model.
2. Trained the LSI model using the TF-IDF representation of the corpus.
3. Created a similarity index for efficient query processing.

Results:

- Evaluated the LSI model's performance using Mean Average Precision (MAP).
- Results showed a significant improvement in retrieving relevant documents compared to the baseline methods.

Experiment 2: Deep Learning Approach

Goal: Compare the LSI model with a deep learning-based semantic search approach.

Steps:

1. Used the Sentence-BERT (LaBSE) model to encode the corpus and queries.
2. Implemented semantic search using cosine similarity on the encoded embeddings.
3. Evaluated the deep learning model's performance and compared it to the LSI model.

Results:

- The LaBSE model demonstrated superior performance in understanding the semantic meaning of queries and documents.
- Provided a more nuanced retrieval of relevant documents compared to the traditional LSI model.

Overall Conclusion

The project successfully demonstrated the effectiveness of both traditional NLP methods and deep learning approaches for semantic search. The deep learning model (LaBSE) outperformed the traditional LSI model in terms of relevance and accuracy.

Tools and Libraries Used

NumPy: For numerical operations.

pandas: For data manipulation and analysis.

spaCy: For tokenization and preprocessing.

Gensim: For topic modeling and similarity computations.

Matplotlib: For visualization.

WordCloud: For generating word clouds.

Sentence-Transformers: For deep learning-based semantic search.

External Resources

Kaggle: Movies Similarity

Resource: Semantic Search using NLP

Reflection Questions

What was the biggest challenge you faced during this project?

The biggest challenge was ensuring the preprocessing steps effectively removed noise while retaining meaningful information.

What do you think you have learned from the project?

I learned how to build feature extraction models, and implement a semantic search system.