Movie Finder

Group: 08

Patrick Eckel, Marcus Gugacs, Martin Tobias Klug, Lukas Leitner

Introduction

Motivation

- Lot’s of video content online
- Many different streaming providers
- Central place for content curation
- Value users time
- Personalized recommendations

Introduction

Research Question

- Can we build a central system which provides recommendations of various streaming services to effectively reduce the users effort of finding content?

Data

Movie Dataset

- Original Columns:
  - id, title, genres, original language, overview, popularity, production companies, release date, budget, revenue, runtime, status, tagline, vote average, vote count, credits, keywords, poster path, backdrop path, recommendations
- Reduced to:
  - id, title, genres, original language, overview, popularity, vote average, credits, keywords, poster path, release year
- Added column: rich features

Data

Subtitles

- Subtitles provided by API (Key required)
- Download / processed on demand
- Raw subtitles
- Preprocessed by removing:
  - timestamps, ids, html tags/entities, parentheses, brackets, braces, musical notes, metadata, speakers, empty lines

Methods

Sequence Transformer

- Model: sentence-transformers/all-mpnet-base-v2
- Semantic text embedding
  - Used to similarity between user query and movie features
- MPNet allows for dense vector representation
  - optimal for semantic sentence similarity
- Processing chunks of max 512 Tokens

Methods

Emotion Classifier

- Model: j-hartmann/emotion-english-distilroberta-base
- Based on DistilRoBERTa
- Classify emotions in english text
  - Supports: Anger, disgust, fear, joy, neutral, sadness, surprise
- Mapping user mood preference to support emotions
- Measure alignment

Methods

TF-IDF Vectorization

- Generate vector representation of text (scikit)
- Enable similarity matching
- Required text preprocessing:
  - Lemmatization (WordNetLemmatizer)
  - Stop word removal (StopWords)
  - Special character cleaning
  - Case normalization
  - Minimum token length

Methods

Movie Introduction Summarization

- Model: facebook/bart-large-cnn
  - Summarization pipeline
  - Based on BART
- Purpose:
  - Creates introductory summary from pre-processed movie subtitles
  - Uses first chunk (1024 tokens) of subtitles for better performance and to avoid spoilers

Methods

Keyword Extraction

- Model: KeyBERT
  - Based on BERT embeddings (unsupervised)
  - Semantic similarity for ranking
- Purpose:
  - Extracts key themes from cleaned movie subtitles
  - Returns top 3 keywords / key themes

System

Overview

1. Initial Filtering
    1.1. Language
    1.2. Era (release year timespan)
    1.3. Genre
    1.4. Minimum popularity
    1.5. Minimum vote average

System
Overview

2. Feature Processing

    2.1. Load cached semantic embeddings or compute them

    2.2. Generate TF-IDF Matrix

    2.3. Encode query text (combined user input)

    2.4. Calculate emotion alignment score

System
Overview

3. Semantic Computation

    3.1. Cosine similarity of semantic

    3.2. TF-IDF cosine similarity

    3.3. Emotional <-> Mood alignment score

    3.4. Weighted score computation

Results

Analysis 1/2

- Internal team evaluation
- Standardized questionnaire
- Repeated evaluation (3 Runs)
- Gathered values:
  - Averaged
  - Visualization
  - Interpretation
  - Discussion

Results

Interpretation 1/2

Results

Analysis 2/2

- Python script (objective evaluation)
- Random test cases and metrics
-

Results

Interpretation 1/2

Live Demo

Conclusion

- Usable and efficient recommendations
- Tweaking and fine-tuning
- Minor tweaks lead to significant changes

Questions?