16:954:577:01 Statistical Software Final Project Report

Movie Recommendation System Using Transformers

#Team 19:

1. Arvind Chary Padala (ap2522)
2. Ram Sampreeth Budireddy (rb1424)
3. Tanishq Sharma (ts1266)

Introduction

Modern movie recommendation systems often rely on user reviews and ratings but struggle with biases, sparse data, and a lack of personalization for conceptual queries. This project aims to overcome these challenges by building a recommendation system that moves away from reliance on reviews and ratings. Instead, it focuses on movie plot summaries and user-provided descriptions, enabling the system to identify movies aligned with the user's narrative and thematic preferences. This approach makes the system more accessible and unbiased, especially for users who wish to explore beyond popular consensus.

The core innovation of our system lies in how the problem is divided into three interconnected NLP tasks, each addressing a specific aspect of movie recommendation:

1. **Genre Classification:** The system uses **DistilBERT** to classify user descriptions into genres, narrowing the search space and aligning recommendations with the user's thematic preferences. For example, if a user describes a desire for a "dark thriller with twists," the model classifies this as a thriller genre and further refines recommendations within this category. The genre classification task was a **multinomial classification problem**, where each movie could belong to multiple genres from 15 predefined categories. Metrics like accuracy and F1-score were used to evaluate performance across all genres.
2. **Semantic Matching:** After identifying the genre, the system leverages plot summaries to match the narrative structure and themes described by the user query. Using dense embeddings generated from fine-tuned transformer models, creates a more nuanced understanding of the query and plots, going beyond keyword matching. This leverages the power of sentence-BERT

for precise semantic understanding and FAISS for fast embedding retrieval, ensuring that the system delivers accurate and efficient movie recommendations.

3. **Question Answering:** To enhance user satisfaction, the system provides detailed answers to questions about the recommended movies, such as information about the cast, director, or storyline elements. This functionality ensures the system is not only accurate in recommendations but also interactive and informative.

For the baseline, we implemented **Logistic Regression** and **Naive Bayes** models using **TF-IDF vectorization** to represent plot summaries. While these models provided a starting point, achieving an accuracy of ~29%, they struggled to capture the semantic nuances of movie plots. To overcome this limitation, we used **DistilBERT**, a transformer-based architecture, fine-tuned specifically for genre classification. This transition to transformers significantly improved performance, with the fine-tuned model achieving an accuracy of 61% and better generalization across genres.

Data Collection and Preprocessing

1. Data Collection

The foundation of our movie recommendation system is the CMU Movie Summary Corpus, a comprehensive dataset developed by researchers at Carnegie Mellon University. This dataset includes **42,306 movies**, each with detailed plot summaries and associated metadata. Key highlights of the dataset include:

- **Plot Summaries:** Concise descriptions of movie narratives, sourced from Wikipedia. These summaries form the primary textual data used to build embeddings and perform semantic matching.
- **Genres:** Multiple genre labels per movie.
- **Release date:** Dates of movie releases spanning decades.
- **Language:** Languages spoken in the movies.
- **Box Office Revenue:** Financial performance metrics.
- **Character Data:** For certain movies, character metadata is available, detailing actor demographics (age, gender, ethnicity, etc.) and other role-specific information.
- **Supplementary Processed Data:** The dataset includes Stanford CoreNLP-processed summaries, providing additional linguistic features such as tokenization, named entity recognition (NER), and coreference resolution.

This rich dataset allows us to build a robust system capable of leveraging narrative and contextual information to make accurate recommendations and answer user queries.

	id	plot	title	genre
0	23890098	shlykov a hardworking taxi driver and lyosha a...	Taxi Blues	[Drama, World cinema]
1	31186339	the nation of panem consists of a wealthy capi...	The Hunger Games	[Action/Adventure, Science Fiction, Action, Dr...
2	20663735	poovalli induchoodan is sentenced for six year...	Narasimham	[Musical, Action, Drama, Bollywood]
3	2231378	the lemon drop kid a new york city swindler is...	The Lemon Drop Kid	[Screwball comedy, Comedy]
4	595909	seventhday adventist church pastor michael cha...	A Cry in the Dark	[Crime Fiction, Drama, Docudrama, World cinema...

Fig 1: Plot Summary Data

	id	title	genres	plot_summary	Release Year	Title	Origin/Ethnicity	Director	Cast	Genre
0	31186339	the hunger games	[Action/Adventure, Science Fiction, Action, Dr...	the nation of panem consists of a wealthy capi...	2012	the hunger games	American	Gary Ross	Jennifer Lawrence, Josh Hutcherson, Liam Hemsw...	action drama, science fiction
1	20663735	narasimham	[Musical, Action, Drama, Bollywood]	poovalli induchoodan is sentenced for six yea...	2000	narasimham	Malayalam	Shaji Kailas	Mohanlal, Aiswarya	unknown

Fig 2: Movie Metadata

2. Data Preprocessing

To prepare the dataset for use in our recommendation pipeline, several preprocessing steps were applied:

1. **Removing Noise:** Non-informative characters, extra spaces, and special symbols (e.g., HTML tags or escape characters) were removed from plot summaries.
2. **Text Normalization:** Converted all text to lowercase to maintain uniformity during tokenization.
3. **Genre Labels:** Movies were mapped to their respective genres using metadata. Since genres could overlap across movies, the labels were encoded for multi-class classification.
4. **Missing or Ambiguous Values:** Missing metadata values, such as undefined runtimes or release dates, were either replaced with median values (numerical) or ignored if non-essential.
5. **Genre Distribution:** Some genres (e.g., "Drama") were heavily represented, while others (e.g., "Cult" or "Musical") were sparse. To address this:
    a. Class weights were calculated and incorporated into the loss function during genre classification training.

b. Data augmentation techniques, such as oversampling underrepresented classes, were applied.

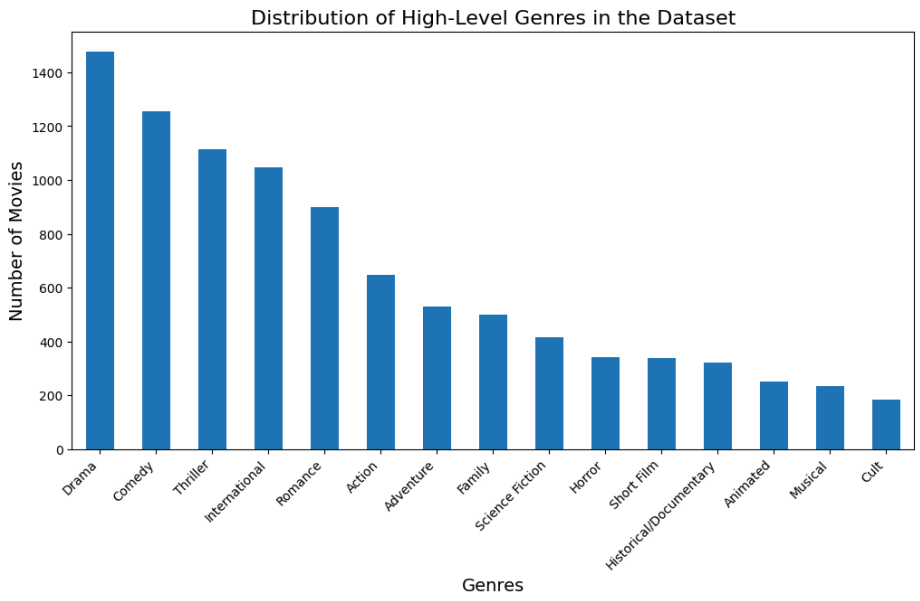

Fig 3: Genre Distribution

6. **QA Dataset:** A synthetic question-answering dataset was created by combining plot summaries and metadata to train the question-answering model. Relevant metadata fields such as actors, directors, runtime, and genres were merged with the plot summaries to form a unified context, and question-answer pairs were generated from this context.

	question	answer	context
135133	Who is the director of children of the revolut...	Peter Duncan	Title: children of the revolution. Plot: Joan ...
65426	Who are the main actors in the shrink is in?	Courteney Cox, David Arquette	Title: the shrink is in. Plot: Samantha (Court...
21674	Who are the main actors in bluff master?	Shammi Kapoor, Saira Banu, Pran, Lalita Pawar,...	Title: bluff master. Plot: Ashok (Shammi Kapoo...
7917	What are the genres of road train?	['Horror']	Title: road train. Plot: Marcus, his best frie...
14730	What is the plot of mandingo?	The movie is set in the Deep South of the Unit...	Title: mandingo. Plot: The movie is set in the...

Fig 4: A sample from the QA dataset

7. **Training and Validation:** The dataset was split into training (70%), validation (20%), and test(10%). Stratified splitting ensured a balanced representation of genres across all subsets.
8. **Data Augmentation:** Synonym substitution and paraphrasing techniques were applied to plot summaries to simulate diverse narrative structures and improve model robustness.

By performing these preprocessing steps, we ensured that the data was clean, structured, and ready to be utilized in the multi-task pipeline for genre classification, semantic matching, and question answering. This comprehensive preparation allowed the system to effectively interpret user queries and provide accurate, personalized recommendations.

Model Pipeline and Experiments

1. Genre Classification

Task: Predict movie genres based on user queries or plot summaries.

Baseline Models: Logistic Regression and Naïve Bayes using TF-IDF features.

Transformer Model: Fine-tuned DistilBERT for multi-class classification, mapping plot summaries to genres.

Hyperparameters: Learning rate = 5e-5, Batch size = 32, Epochs = 3.

Challenges and Optimizations:

- **Class Imbalance:** Addressed using weight balancing.
- **Overfitting:** Implemented early stopping based on validation loss.
- **Training Stability:** A linear decay learning rate scheduler with warm-up steps was used.

The Genre Classification process begins with movie plot summaries, enhanced with metadata such as genres, language, and release year. These inputs are tokenized using the DistilBERT tokenizer and processed through a fine-tuned DistilBERT model (distilbert-base-uncased). The task is framed as a multinomial classification problem across 15 predefined genres. Key parameters include a learning rate of 3e-5, a batch size of 16, and a sequence length of 256 tokens to handle longer summaries. To address the class imbalance, the weighted cross-entropy loss was used to prioritize underrepresented genres. The model was trained over three epochs with a linear learning rate scheduler for stability.

Performance was evaluated using accuracy, precision, recall, and F1-score, with comparisons to baseline models like Logistic Regression and Naive Bayes, which achieved ~29% accuracy. In contrast, the fine-tuned DistilBERT significantly improved accuracy to 61%, highlighting the effectiveness of transformer-based architectures in capturing semantic nuances for genre classification. This robust setup laid the foundation for delivering precise and personalized movie recommendations. By tuning parameters such as token length, and batch size the model's performance can be further improved.

The table below compares the key metrics across different models, showcasing the progression from baseline methods to the fine-tuned transformer model.

Model	Precision	Recall	Accuracy	F1-Score	Remarks
Naive Bayes	0.14	0.22	21.60%	0.15	Struggled with semantic nuances and genre overlap.
Logistic Regression	0.19	0.18	17.89%	0.17	Limited ability to capture complex narrative relationships.
Neural Network	0.16	0.17	16.79%	0.16	Performed poorly without task-specific tuning.
DistilBERT (Pre-trained)	0.35	0.44	44.00%	0.35	Captured contextual relationships but needed fine-tuning.
DistilBERT (Fine-Tuned)	0.67	0.61	61.15%	0.62	Significantly improved accuracy and minority class performance.

Table 1: Key Metrics Comparision Table

2. Semantic Matching

Task: Recommend movies by matching user queries and classified genres with movie plot summaries.

Baseline Model: TF-IDF vectorization and cosine similarity.

Transformer Model: Utilized pre-trained Sentence BERT(**all-MiniLM-L6-v2**) to generate dense vector embeddings for user queries and plot summaries.

Query Embeddings: Concatenated user queries with predicted genres to enhance contextual understanding.

Recommendation Method: Retrieved top-3 movie recommendations based on cosine similarity of embeddings.

Evaluation: Sentence BERT embeddings demonstrated higher semantic matching accuracy compared to TF-IDF-based methods.

The sentence-BERT model (all-MiniLM-L6-v2) was employed to generate semantic embeddings for all the plot summaries in the dataset. This model is particularly well-suited for such tasks because it is compact, computationally efficient, and produces fixed-length, dense embeddings (384 dimensions) that effectively encapsulate the semantic meaning of text, irrespective of its length. The generated embeddings were stored using FAISS (Facebook AI Similarity Search), a library specifically designed for efficient similarity search and clustering of dense vectors. FAISS is highly optimized for searching in large collections of embeddings, providing scalable indexing methods and fast nearest-neighbor search capabilities.

When a user provides a query describing the type of movie they want to watch, the query is first concatenated with the genre predicted by the genre classification model to enhance contextual understanding. The sentence-BERT model is then applied to the updated query to generate its semantic embedding. To recommend movies, cosine similarity is computed between the query embedding and the pre-stored embeddings in FAISS. Based on these similarity scores, the top 3 movies that align most closely with the query's semantic meaning are selected and recommended to the user.

Compared to TF-IDF, which is our baseline model, sentence-BERT embeddings capture semantic meaning, enabling context-aware matching for more nuanced tasks. They are also dense and fixed-length, reducing memory usage and improving retrieval speed over TF-IDF's sparse, high-dimensional vectors.

3. Question Answering

Task: Answer user questions about recommended movies using metadata and plot summaries as context. (Extractive question answering)

Transformer Model: Fine-tuned DistilBERTForQuestionAnswering, a DistilBERT variant pre-trained on SQuAD (Stanford Question Answering Dataset).

Hyperparameters: Learning rate = 5e-5, Batch size = 16, Epochs = 3.

Challenges:

- **Text Truncation:** Long plot summaries were truncated, potentially losing crucial context.
- **Computation Costs:** Fine-tuning transformer models required significant time and resources, mitigated by using DistilBERT (lightweight).

In this part of the model, the synthetic question-answering (QA) dataset is utilized consisting of questions, corresponding contexts, and correct answers. The dataset is tokenized, and Distillibert embeddings are utilized, where both the questions and contexts are processed to extract relevant information. The positions of the answers within the context are identified, which are essential for training the model. Along with the position, the answer's embeddings are utilized for contextual significance. The dataset is split into training and validation sets, and a pre-trained DistilBERT model for question-answering is fine-tuned on the training data. The model is trained over three epochs using a batch size of 16 and a learning rate of 5e-5. During each epoch, the model's performance is evaluated on the validation set, and the average training and validation losses are reported to track the model's progress. After fine-tuning, the model and tokenizer are saved for future use. For answering new questions, the model is provided with a question and context, and it identifies the most likely answer from within the context. This process

enables the model to learn from the synthetic dataset and effectively answer questions based on new, unseen queries.

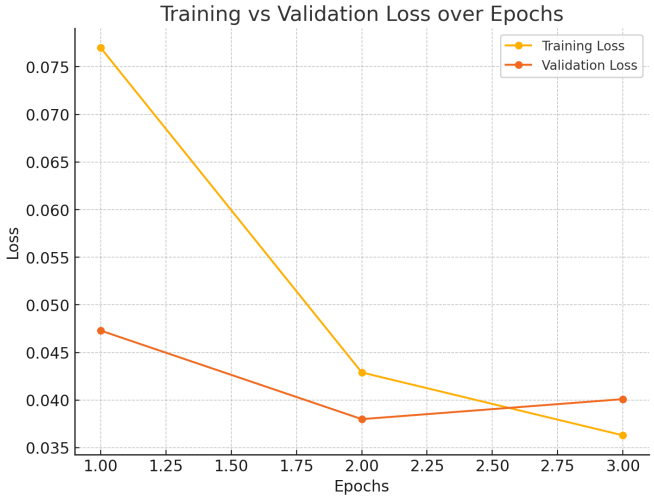

Fig 5: Training vs Validation loss(QA model)

Evaluation Metrics	
Exact Match	0.7457
Recall	0.8160
Precision	0.7457
F1 Score	0.7992

Table 2: Evaluation Metrics(QA model)

The model shows steady improvement with a decreasing training loss, indicating effective learning. However, the slight increase in validation loss towards the end suggests some overfitting. Based on the validation loss curve, the weights from Epoch 2 were saved. The evaluation metrics highlight a solid performance, with an F1 score of 0.7992 reflecting a good balance between precision (0.7457) and recall (0.8160). The higher recall indicates the model is good at identifying relevant instances, while the slightly lower precision suggests room for improvement in minimizing false positives. Overall, the model is effective but could benefit from larger models like BERT also, an increased input sequence length would enhance this model to help generalize better.

Conclusion:

The integration of these three models creates a seamless and effective recommendation system. Each model contributes uniquely to the overall experience, Genre Classification ensures that user queries are directed towards the most relevant subset of movies, filtering out irrelevant options and improving recommendation precision. Semantic Matching uses advanced embeddings to identify movies with narratives closely aligned to the user's description, providing personalized and contextually accurate recommendations. Question Answering enhances interactivity, allowing users to ask detailed questions about recommended movies. Fine-tuned on concatenated metadata and plots, the QA model provides precise and intuitive answers, enriching the user experience.

By combining these specialized tasks, we learned that we can achieve enhanced semantic similarity, intuitive query handling, and a user-centric interface that aligns with individual preferences.

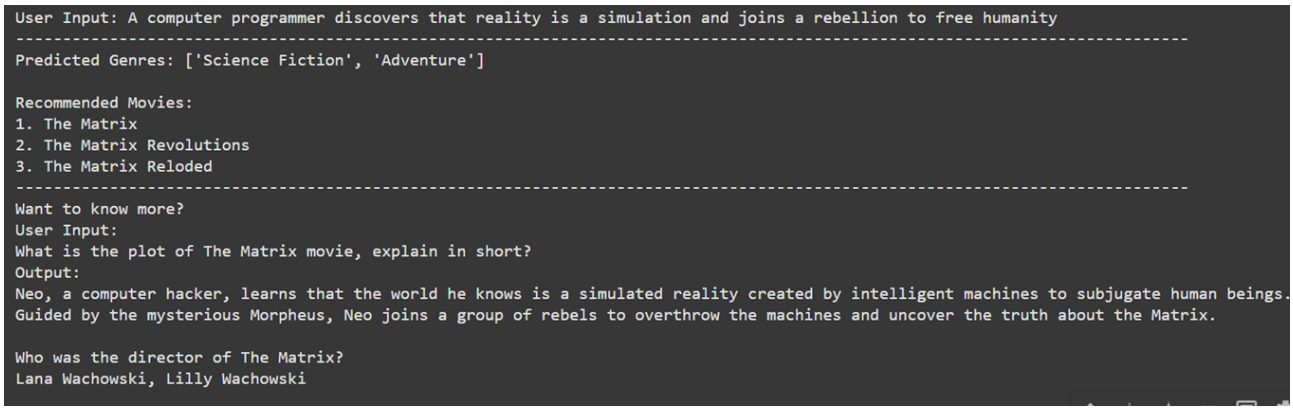

Fig 6: Complete pipeline demo

The models can be further improved by incorporating larger models like BERT for the classification task, also for question answering task, instead of performing an extractive search, we could perform a generative search using models like T5 and BART which have higher computational requirements.

Application of Research Papers in Our Project:

"Multi-Label Classification of Hate Speech Severity on Social Media using BERT Model"

The paper *"Multi-Label Classification of Hate Speech Severity on Social Media using BERT Model"* explores using BERT for the challenging task of identifying and classifying hate speech into multiple severity levels on social media. Hate speech often belongs to multiple overlapping categories, requiring a robust multi-label classification approach. The authors fine-tuned BERT on a dataset annotated with multiple hate speech categories and severity levels, incorporating techniques like weighted loss functions to handle class imbalance. Their model leveraged the semantic understanding capabilities of BERT to outperform traditional machine learning models, achieving significant improvements in precision and recall across all categories. This work highlights the power of pre-trained transformers in capturing the nuanced relationships between text and overlapping labels.

Relevance to Our Project:

This paper's emphasis on multi-label classification using BERT aligns closely with the **genre classification task** in our project. Like hate speech severity categories, movie genres often overlap, requiring the model to predict multiple labels simultaneously. Inspired by this work, we adopted **weighted**

cross-entropy loss to address the class imbalance in our dataset, ensuring underrepresented genres like "Cult" or "Musical" were adequately accounted for. Additionally, the use of **BERT for semantic understanding** informed our decision to fine-tune DistilBERT, a lightweight variant of BERT, for genre classification. This allowed us to capture the contextual nuances of plot summaries, improving classification accuracy significantly over baseline models.

The paper also influenced our **hyperparameter optimization strategy**, particularly regarding token length and class weighting. By demonstrating the importance of tuning token limits for better context capture, we set a maximum sequence length of 256 tokens to handle longer plot summaries effectively. Moreover, the paper reinforced the importance of using precision, recall, and F1-score for evaluation, ensuring our model's performance was robust across all genres, including those with fewer samples. These insights collectively helped us refine our approach, enabling a more effective multi-label classification pipeline for movie genres.

References

Research Papers:

1. B. D. Dirting, G. A. Chukwudebe, E. C. Nwokorie, and I. I. Ayogu, "Multi-Label Classification of Hate Speech Severity on Social Media using BERT Model," *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development*, 2022.
2. Vaswani et al., "Attention is All You Need" (2017) – The seminal paper introducing the transformer architecture.

Models:

1. Genre Classification: https://huggingface.co/distilbert/distilbert-base-uncased
2. Semantic Matching: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
3. QA: https://huggingface.co/distilbert/distilbert-base-uncased-distilled-squad

Dataset:

CMU Movie Summary Corpus (link)