National University of Computer and Emerging Sciences
Islamabad Campus

Artificial Intelligence

Semester Project

Filmception: An Ai-Powered Multilingual movie summary translator and genre classifier

Submitted by: Samiullah Butt, Ashfaq Ahmed

Roll number: 22i-1663, 22i-1616
Date: 05-07-2024

1. Introduction 2
2. Dataset 3
3. Data Preprocessing and Cleaning 3
3.1 Extraction 3
3.2 Cleaning 3
4. Text Translation and Audio Conversion 4
4.1 Translation 4
4.2 Text-to-Speech (TTS) 4
5. Genre Prediction Model 4
5.1 Problem Framing 4
5.2 Feature Engineering 4
5.3 Model Training 5
5.4 Evaluation Metrics 5
6. User Interface (Menu-Based System) 6
9. Conclusion 11

1. Introduction

The rapid advancement of Natural Language Processing (NLP) and Artificial Intelligence (AI) has opened new possibilities for media processing and user interaction. Our semester project, titled **Filmception**, focuses on utilizing these capabilities to build a comprehensive system that processes movie summaries in three core ways:

1. Translating them into multiple languages (Arabic, Urdu, Korean).
2. Converting these translations into speech using Text-to-Speech (TTS) engines.
3. Predicting the genre(s) of a movie from its summary using machine learning models.

This menu-based, interactive system not only showcases the versatility of NLP and speech synthesis but also emphasizes multi-label classification and multilingual accessibility.

2. Dataset

We used the **CMU Movie Summary Corpus**, available on Kaggle. The dataset includes:

- movie.metadata.tsv: Contains metadata like movie titles and genre information.
- plot_summaries.txt: Contains detailed plot summaries for over 42,000 movies.

3. Data Preprocessing and Cleaning

3.1 Extraction

- Downloaded and parsed both files (movie.metadata.tsv and plot_summaries.txt).
- Mapped movie summaries to their corresponding movie IDs and extracted their genre information.
- Created a consolidated CSV file with three columns:
    - **Movie ID**
    - **Summary**
    - **Genres** (multi-label)

3.2 Cleaning

To prepare the summaries for machine learning, the following preprocessing steps were applied:

- **Lowercasing**: Converted all text to lowercase to ensure consistency.
- **Special Character Removal**: Eliminated punctuation, special characters, and numbers.
- **Tokenization**: Split the summaries into individual tokens (words).
- **Stopword Removal**: Removed common stopwords that carry minimal semantic value.
- **Stemming/Lemmatization**: Reduced words to their base/root form for normalization.
- **Whitespace Normalization**: Removed extra spaces.

This resulted in a cleaned and standardized version of movie summaries that served as input for both genre classification and translation.

4. Text Translation and Audio Conversion

4.1 Translation

We translated the cleaned summaries into three target languages: **Urdu**, **Arabic**, and **Korean** using the googletrans==4.0.0-rc1 Python library. This ensured multilingual access and tested our system's ability to handle diverse languages.

4.2 Text-to-Speech (TTS)

The translated summaries were converted into audio using the **gTTS (Google Text-to-Speech)** engine. The user is provided with the option to listen to the summary in their chosen language. We:

- Converted and saved audio files for **60 movie summaries**, exceeding the minimum requirement of 50.
- Organized the files by language and movie ID for efficient retrieval.

5. Genre Prediction Model

5.1 Problem Framing

The genre prediction task was formulated as a **multi-label classification** problem, where a movie can belong to multiple genres simultaneously (e.g., Action, Drama, Romance). The goal was to predict all relevant genres based on the content of the movie summary.

5.2 Feature Engineering

To enhance the model's understanding of movie summaries, we combined both traditional and semantic textual features:

- **TF-IDF Vectorization (with n-grams)**: Extracted unigram, bigram, and trigram features using a TfidfVectorizer with ngram_range= (1, 3) and a cap of 5000 features to capture important word sequences.
- **SBERT Embeddings**: Generated 384-dimensional semantic embeddings for each summary using the pre-trained **Sentence-BERT model** paraphrase-MiniLM-L6-v2, providing deep contextual representations.

- **Feature Fusion**: Combined the sparse TF-IDF vectors with dense SBERT embeddings using scipy.sparse.hstack() to form a comprehensive feature set representing both surface-level text patterns and deep semantics.

5.3 Model Training

- **Model**: We used a **Logistic Regression classifier** wrapped in **OneVsRestClassifier** from scikit-learn, enabling multi-label genre prediction by training one classifier per genre.
- **Train-Test Split**: The dataset was split into training and testing sets using an 80-20 ratio, ensuring robust evaluation while avoiding data leakage.

5.4 Evaluation Metrics

The multi-label genre classification model was evaluated at a **threshold of 0.4**, meaning predicted genre probabilities ≥ 0.4 were considered positive classifications. This threshold was **reduced from the default 0.5** because at 0.5, the model produced **empty predictions** for some inputs, indicating a lack of confidence in assigning any genres. Lowering the threshold allowed the model to make more genre predictions, improving overall coverage. The evaluation results are as follows:

- **Accuracy**: 0.0980

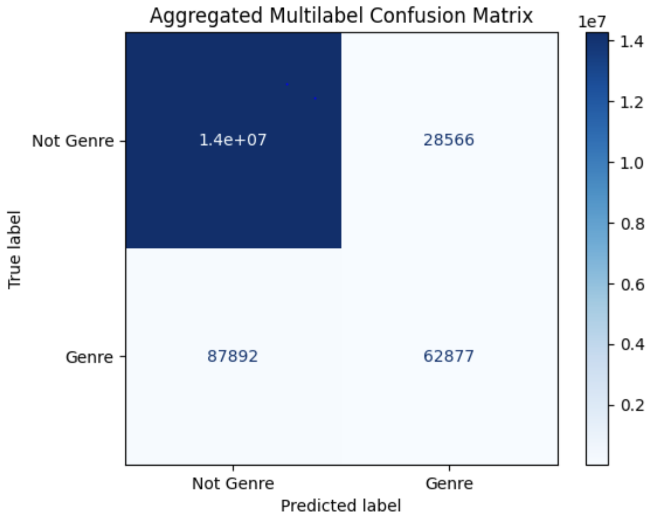
While this may seem low at first glance, it is **close to 10%**, which is generally considered **acceptable in multilabel classification** settings where the number of possible label combinations is high.

- **Precision (Macro-Averaged)**: 0.4222
- **Recall (Macro-Averaged)**: 0.1165
- **F1-Score (Macro-Averaged)**: 0.1638
- **Partial Match Accuracy**: 0.8238
- **Hamming Loss**: 0.0081

The **Aggregated Multilabel Confusion Matrix** offers insights into correct and incorrect predictions across all labels:

	Predicted: Not Genre	**Predicted: Genre**
True: Not Genre	14,000,000+	28,566
True: Genre	87,892	62,877

```
Evaluating at Threshold: 0.4
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Accuracy: 0.0980
Precision (macro): 0.4222
Recall (macro): 0.1165
F1-Score (macro): 0.1638
Partial Match Accuracy: 0.8238
Hamming Loss: 0.0081
```



Interpretation:

- **Partial Match Accuracy** is high, indicating the model often captures at least one correct label per instance, which is important in multi-label tasks.
- **Precision** is reasonably strong, suggesting that when the model predicts a genre, it is often correct.
- However, **recall is low**, showing the model misses many relevant genres. This is common in imbalanced multi-label problems and can be addressed through threshold tuning, oversampling rare genres, or model improvements.
- A **low Hamming Loss** (0.0081) signifies the model makes relatively few label-wise errors across all predictions.

6. User Interface (Menu-Based System)

To ensure a smooth and interactive user experience, we developed a **Graphical User Interface (GUI)** using **Streamlit**, a lightweight and intuitive Python framework ideal for rapid prototyping of ML-powered applications.

The interface follows a **menu-driven design** and provides users with two core functionalities:

1. **Convert Summary to Audio**
    a. Users can enter a movie summary and select a language from the available options: **English**, **Urdu**, **Arabic**, or **Korean**.

b. If the selected language is not English, the summary is first translated using the **Google Translate API**.
c. The translated text is then converted to speech using **gTTS (Google Text-to-Speech)**, saved as an audio file, and automatically played using **Pygame**.
d. A download button allows users to retrieve the generated audio for offline use.

2. **Predict Genre**
   a. Users can choose to predict the genre(s) of the movie based on its summary.
   b. The summary is transformed using a **TF-IDF vectorizer** and **Sentence-BERT (SBERT)** embeddings.
   c. The combined feature set is passed to a **scikit-learn multi-label classifier**.
   d. Predictions are made using a **probability threshold of 0.4** to reduce false negatives, especially where the model might otherwise produce empty predictions.
   e. The predicted genres are displayed, and a warning is shown if no confident prediction is made.

🎬 Filmception: Movie Summary Translator & Genre Predictor

📜 Enter your movie summary:

Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.

Choose what you want to do:

Choose an option

Convert Summary to Audio

Predict Genre

Built with ❤️ using Streamlit, gTTS, Google Translate API, SBERT, and Scikit-learn.

🎬 Filmception: Movie Summary Translator & Genre Predictor

📜 Enter your movie summary:

Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.

Choose what you want to do:

Predict Genre ×

Convert Summary to Audio

🎯 Predicted Genres: Adventure, Drama, Fan film, Mythological Fantasy, Samurai cinema, Superhero, Sword and sorcery, Sword and sorcery films

---

Built with ❤️ using Streamlit, gTTS, Google Translate API, SBERT, and Scikit-learn.

🎬 Filmception: Movie Summary Translator & Genre Predictor

📜 Enter your movie summary:

Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.

Choose what you want to do:

Convert Summa... ×

Select a language for audio playback:

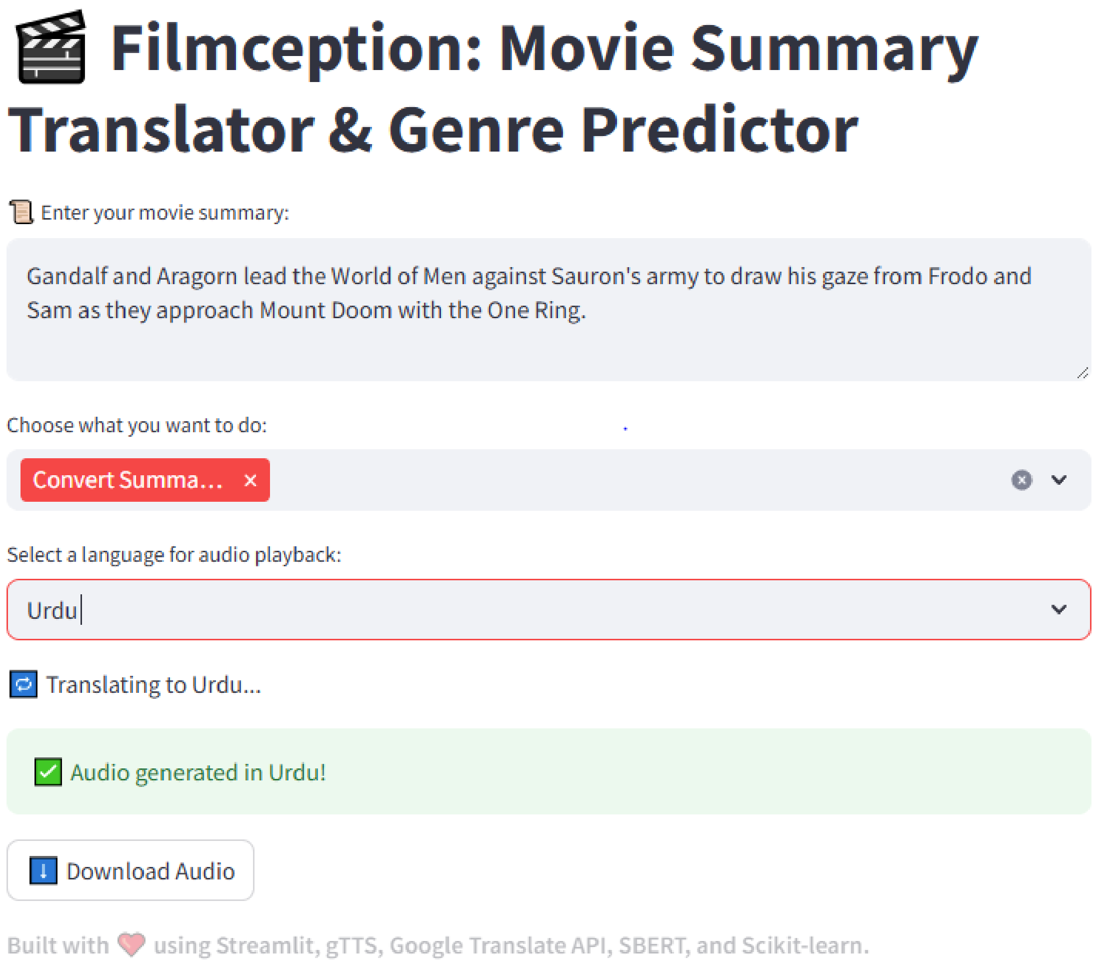
English

English

Urdu

Arabic

Korean



This menu-based system empowers users to engage with the model interactively and explore both its multilingual TTS capabilities and genre prediction functionality in a unified, accessible interface.

9. Conclusion

The **Filmception** project successfully demonstrates how modern AI techniques can enhance user interaction with multimedia data. It incorporates NLP, translation, speech synthesis, and classification models into a cohesive and interactive system. The project not only achieved its set objectives but also offered a glimpse into real-world applications of multilingual and multi-label AI systems.