Report

1. **Methodology for determining movie genre label codebook**

   I initiated the process by identifying the labels for the films on IMDb and TMDB and investigating the APIs available for labeling the dataset. Subsequently, I discovered pre-existing labels for this dataset and integrated them into the dataset. Each trailer was associated with multiple labels.

   The challenge at hand pertained to the implementation of multi-modality, incorporating both the video and audio signals of the trailer. Initially, I endeavored to utilize the VGG16 model to extract the features. However, due to time constraints, I found that the computational demands of this approach were considerable. Consequently, I opted to prioritize the audio component of the analysis, recognizing the necessity of expeditious decision-making within the allotted timeframe.

   The subsequent stage involved the conversion of all YouTube links into textual data. Initially, I attempted to accomplish this by downloading the audio from the videos and subsequently applying the Whisper Speech-to-Text model for transcription. Regrettably, I encountered significant delays in this process, as each video required a considerable amount of time for transcription.

   Subsequently, I resorted to utilizing the YouTube API to download the English subtitles of the trailers. However, this approach encountered a setback as numerous trailers lacked available subtitles, thereby limiting the efficacy of this method.

   I successfully transcribed 4,356 records from the available trailers. However, it's worth noting that a few trailers were missing, and for some links, subtitles were unavailable. As a result, the dataset formed exhibits a multiclass multi-label structure, encompassing 19 unique genre classes.

2. **Partitioning methodology**

   We partitioned the dataset into training and testing subsets using an 80-20 split ratio, where 80% of the data was allocated for training purposes, and the remaining 20% was reserved for testing.

3. **Modelling approach**

   a. **ML Based**

      As a preprocessing step, I eliminated all stop words from the transcriptions. Subsequently, I employed tokenization and lemmatization techniques on the text data. Following this, I utilized the TfidfVectorizer to generate a vector representation for each sentence in the dataset

      Following preprocessing, I proceeded to train a logistic regression model using the MultiOutputClassifier framework. This allowed for the prediction of multiple classes simultaneously, accommodating the multi-label nature of the dataset.

      Experimental protocol and performance metric calculation

      After training the logistic regression model with the MultiOutputClassifier, I evaluated its performance using the accuracy_score metric. This metric provides insight into the overall accuracy of the model's predictions across all classes in the dataset.

      How to execute this

Got it, it seems you have a structured setup for your machine learning project. Here's how you can proceed:

First, ensure you have the required dependencies installed by using the `req.txt` file.
`train.ipynb` notebook contains the training code
`test.ipynb` notebook contains the test coode
You can also test the model from the command line using the `test.py` script. As you mentioned, you can run:
python test.py youtube_link
Replace `youtube_link` with the actual YouTube link you want to test.

b. **DL Based**
Fine-Tuning DistilBERT for Movie Genre Prediction
I employed the DistilBERT (Victor Sanh, 2019) model from Hugging Face and conducted fine-tuning to adapt it to the task of predicting movie genres based on textual data.
DistilBERT, a distilled version of the BERT model, was chosen due to its effectiveness in natural language processing tasks. Pre-trained weights were obtained from the Hugging Face model repository, providing a strong foundation for further adaptation.
A batch size of 16 was selected for training. Training was conducted over multiple epochs to iteratively update the model's parameters. Initially, two epochs were executed with the pre-existing weights frozen. This facilitated gradual adaptation to the dataset without perturbing the pre-trained representations significantly. Subsequently, an additional seven epochs were performed after unfreezing the layers. This enabled the model to learn more intricately from the dataset, leveraging both the pre-trained and task-specific information.
Overall, the fine-tuning process adhered to established best practices in transfer learning and model adaptation. By leveraging the powerful representations learned by DistilBERT in conjunction with task-specific data, we aimed to develop a robust and effective model for movie genre prediction.

Bibliography

Victor Sanh, L. D. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.