Report

Name : Akhilesh Soni
College Name : International Institute of Information Technology , Hyderabad , India

Platform : Python(2.7) + some dependencies,libraries (NLTK toolkit , Word2Vec)

Analysing Dataset :
We are given 2601 examples which have following attributes :
• asset_id: identifier of the movie;
• title: the title of the movie;
• summary: a short text describing the movie;
• poster_movie_url: it is the url location of the movie poster;
• poster_trailer_url: if available, it is the url location of the movie trailer.
• Moreover, we have 17 binary attributes assign the movie to a genre. The same movie can belong to multiple genres simultaneously.

Training Data = 80%
Testing Data = 20%

Classified Label Prediction : For predicting the output label a threshold was set and based on that we predict the genre label.

Evaluation Metric Used : F1 score is used as an evaluation metric which is harmonic mean of precision and recall.
Here,
Precision = True Positive / (True Positive + False Positive)
Recall = True Positive / (True Positive + False Negative)
F1 score = 2*Precision*Recall/(Precision + Recall)

Method 1 : This approach is based on working with only summaries of each movie and extract features and convert it into a feature vector.
In this approach I used pretrained Glove Embeddings trained on Wikipedia Italian Corpus . Then the steps involved were:

- Tokenization of Summary text
- Removal of Stop Words
- Now , use Glove Embeddings which gives featurised representation of each word (300 dimension)which is then used as a feature vector.
- Now, mean average of these embeddings is taken according to frequency of each word occuring in the summary.

Method 2 : This approach is based on working with only images of each movie poster and extract features and convert it into a feature vector.

In this approach I used pretrained VGG feature extractor(pretrained model from imagenet data) and obtained a featurised representation of each image . Each feature vector is of 8192 dimension. Then these features are fed into the neural network to train our model.

Method 3 : This approach is slight modification of 2nd approach and now I extract featurised representation of both move_poster image and movie_trailer image.

In this approach I used pretrained VGG feature extractor and obtained a featurised representation of each image . Finally our feature vector is mean of feature vector_1(movie_poster image) and vector_2(movie_trailer image).This gives slightly better results as compared to 2nd approach.

Method 4 : This approach is based on concatenating features(both visual and text) and then this feature vector is used to train our model.
Feature vectors obtained individually based on approach 1 and approach 4 are concatenated and then these features are fed into neural network to train our model.

Comparison of Different Approaches :

Approach	Accuracy(%)	Precision	Recall	F1 Score
Method 1	**89.07**	**0.6191**	**0.6062**	**0.6126**
Method 2	**92.54**	**0.4034**	**0.4186**	**0.4108**
Method 3	**92.61**	**0.4540**	**0.4399**	**0.4468**
Method 4	**92.47**	**0.4319**	**0.4982**	**0.4626**

Neural Network Architecture :
3 layer neural network architecture was used in which the input layer , hidden layer has “relu” activation function and at output layer “softmax” activation function was used which gives probability of occurrence of each genre.

References :

- https://medium.com/@franky07724_57962/using-keras-pre-trained-models-for-feature-extraction-in-image-clustering-a142c6cdf5b1
- http://hlt.isti.cnr.it/wordembeddings/