**Movie Genre Prediction and Suggestion Model**

BAN 676 – Deep Learning  
Abhay Bilapatte, Tarun Surapaneni, Tejaswi Kadam

**BAN 676: Deep Learning**  
  
**Sri Sai Tarun Surapaneni**

**Introduction:**

Movies, TV shows and Web Series are the greatest forms of entertainment in today’s era but not everyone has the same taste in terms of genres which they like. Ten years ago, it was very difficult to classify the movie genre but with the help of advanced techniques of ‘Deep Learning’ now it seems very easy. The idea of this project was to classify the user input into one or more movie genres and then suggest few movies based on those genres. In this project I explored the possibility of using Deep Neural Networks, NLP and HAN to classify the keywords as per genre using the movie summary plots and then provide the user with movie names which belong to that genre. Two models were compared in order to testify the working of this project. This module will be useful in saving users time and effort while searching for a specific movie and Deep Learning makes this process convenient and hassle free.

**Keywords:** Deep Learning, NLP, Hierarchical Attention Network (HAN), Genre Classification, Movie Suggestion

**Data Description and Preparation:**

I used the dataset from 'CMU Movie Summary Corpus’ which has links to a dataset of movie plot summaries and associated metadata. The data is merged to create a combined and cleaned dataset with 4 to 5 columns which involve Movie id, Movie Name, Genre and Plot. The dataset consists of 363 different genres, which is then shortened down to 10 major genres. Finally, the dataset is of 36050 rows of movie names which are classified in 10 genres. For development of the environment used is ‘Google Colab Pro’ (premium version) with GPU’s and TPU’s. Since the data is heavily loaded for the training process, this environment is essential for running the code smoothly with or without interruptions.

**Model Description:**

Implemented two approaches for the project, one was the Basic model with LSTM attention mechanism and the second was Hierarchical Attention Network. HANs consist of stacked recurrent neural networks on word level followed by an attention model to extract important to the classification of the sentence and aggregate the representation of those informative words to form a sentence vector. It includes two levels of attention mechanisms — one at the word level and Multi-Label Movie Genre Classification from Plot Summaries, one at the sentence level — that let the model pay more or less attention to individual words and sentences when constructing the representation of the document. The plot summary serves as a keyword supplier and when a user enters a plot genre is predicted and movies are suggested accordingly.

Model Analysis, Optimization and Implementation:

HAN Model Architecture:

Consider HAN as two separate models, First is the Encoder model which involves a conventional RNN (Recurrent Neural Networks) built with Bi-directional LSTM which is then followed by a same sized attention layer. Word encodings are taken as inputs and encoded representation of words is done based on the number of layers in BLSTM, this output is then given to Time Distributed layer (Responsible for running a copy of encoder at each input sequence), this is again followed by BLSTM layer and an attention layer of same size. Finally, used Sigmoid as my output layer.

**Model Type:**

a) Basic Model:

|  |  |
| --- | --- |
| **Parameters** | **Basic Model** |
| Layers | 3 |
| Neurons | LSTM=100, Dense=10, Conv1D=64 |
| Activation Function | Sigmoid |
| Optimization Algorithm | Adam |
| Batch Size | 64 |
| Epochs | 35 |

b) HAN Model:

|  |  |
| --- | --- |
| **Parameters** | **HAN Model** |
| Layers | Word Level – 2 layers & Sentence level- 3 layers |
| Neurons | Word level🡪 LSTM=150, Dense=200,  Sentence Level--> LSTM=150, Dense=200, Dense=10 |
| Activation Function | Sigmoid |
| Optimization Algorithm | Adam |
| Batch Size | 64 |
| Epochs | 30 |

**Model Structure:**

Graphical user interface, application, Teams

Description automatically generated

* There are two levels in this architecture of HAN namely Encoder and Decoder.
* The word embeddings for each word in a sentence are fed into a bi-directional GRU for building a sentence representation, thus creating contextually thorough annotations for each word. Within the first attention layer, each annotation is transformed via a tanh activated fully connected layer.
* Normalized importance weight for each word is taken through word vector by using sigmoid function. Based on these normalized weights, I can compute the sentence vector as a weighted sum of the word annotations. The same technique is repeated for rest of the parts of the plot summary.

**Implementation of the Project:**

Data Preprocessing:

* Since the dataset was huge, preprocessing was a tedious task and hence was subdivided into 4 major parts as mentioned in the above figure.
* Reading the dataset involved reading rows and columns and understanding their significance to my best understanding.
* Since I had two different datasets one for movie id, movie name and its genre and other for summary plots, I had to merge these two datasets to form one single data frame which had 3 important columns i.e. movie id (only considered for data merging), movie name, movie genre and summary plot.
* The most important aspect of preprocessing was Data cleaning, in this case I had a very few missing values in the dataset and had to remove these null values from both rows and columns.
* Consolidating the genre labels was also a part of the cleaning process, where I reduced the 363 genres into 10 major categories.
* Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words and lemmatization function was used to perform this step.
* For binarizing the labels, I used multilabel binarizer which divided the labels in 0 and 1
* Tokenization was also used in the preprocessing step; I have used the GloVeembedding to tokenize sentences into different words. It is an unsupervised learning algorithm for obtaining vector representations for words. Glove is used to compute an index mapping words to known embeddings, by parsing the data dump of pre-trained embeddings.
* I iterated over tokenized words and create dictionaries that keep track of number of tokens, length of sentences, and sentences per plot summary.

**Basic Model Implementation:**

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* The base model involves 2 different approaches, one which has Conv1D with a LSTM layer topped onto it, whose activation function is Sigmoid and Adam is the optimizer used.
* Ran this for 35 epochs with a batch size of 64 and the results obtained after the compilation were not up to the mark as the accuracy was low.
* Changing the Word embedding model to Word2Vec instead of Glove Embedding will probably help with the training part and results will probably improve.

**HAN Model Implementation:**

**Text

Description automatically generated**

* The HAN (Hierarchical Attention Network) model has two parts to it as mentioned above, it’s run at two different levels in Word Level and in Sentence Level.
* The Word level model has 2 layers, one LSTM and another Dense layer.
* The number of neurons in these 2 layers are 150 and 200 respectively.
* For the sentence level model, there are 3 layers LSTM and 2 Dense layers where in one is for output, which is used along with sigmoid activation function and Adam optimizer.
* Time distributed layer is a core part of HAN, it runs a copy of encoder on each i/p sequence, which is followed by BLSTM & an attention layer of same size.
* 30 epochs for this model to see if the model was training properly, the results were good as the model was able to predict the genre of the movie and was also able to suggest few movies related to that genre.
* The training accuracy of this model was 60%.

**Results:**

1. **Basic Model:**

**Graphical user interface, text, application

Description automatically generated**

Graphical user interface, text, application

Description automatically generated**Graphical user interface, text, application

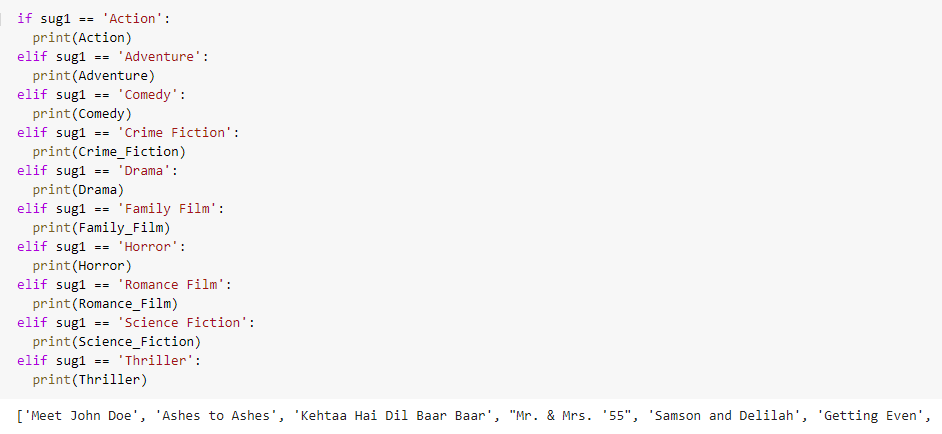
Description automatically generated**

* Above mentioned are the results of the Basic model implementation, in the first image I have the actual genre of a movie based on it’s keywords which belong to variable sug1 as Romance film but with both the basic models, Conv1D+LSTM which predicted the genre to be Crime Fiction and with Glove embedding applied to this basic model, it predicts the genre to be Drama.
* Both the results generated are not correct and need improvements, which can be achieved by training the model and dataset more accurately and using replacements for the word embedding models and I can try changing batch sizes and run the model for more number of epochs with different optimizer and activation function.

1. **Advanced Model: HAN**

**Graphical user interface, text, application

Description automatically generated**

****

* Above mentioned are the results of the Advanced model i.e. HAN, with the variable sug1, the movie genre being predicted is romance film, which is a correct prediction and based on this prediction there are certain romance movies suggested to the user (shown in image 2).
* However, the training accuracy of this model is also not very high and comes around 60% , certain methods with which can increase the accuracy is Train the data in a more accurate manner and try reducing the dataset size so that training is done precisely, increasing the batch size and training the model with more number of epochs can be considered to see if the accuracy increases.

**Future Work:**

* This type of a model where user can get the movie genre just by typing few keyword/s is not widely available in lower rates and hence this model can be used by people to save their time in surfing different entertainment platforms and could get good suggestions on the content which they wish to watch.
* Having said that this project needs future work in the form new and revised models being tested for even more appropriate predictions and suggestions.
* Using different word embedding technique instead of Glove embedding is one part of it.
* Creating my own embedding layer too would be useful in order to see if the performance improves
* Using some other models which have same working functionality as HAN will be a good idea to see if the results vary and if yes by how much margin. For ex. Universal Sentence Encoding (USE Model) along with HAN.

**Conclusion:**

* Achieved significantly more accurate predictions using HAN as compared to the Basic (ConV1D along with LSTM) model.
* This shows that for classification and prediction purposes HAN is a better fit as compared to the basic model.
* Even though I could see improvements in Predictions and Suggestions using HAN model, none of these models where clear winners and will still need some finetuning as mentioned above.

**References:**

* [**https://medium.com/analytics-vidhya/hierarchical-attention-networks-d220318cf87e**](https://medium.com/analytics-vidhya/hierarchical-attention-networks-d220318cf87e)
* [**https://www.researchgate.net/publication/322929271\_Movie\_Genre\_Classification\_from\_Plot\_Summaries\_Using\_Bidirectional\_LSTM**](https://www.researchgate.net/publication/322929271_Movie_Genre_Classification_from_Plot_Summaries_Using_Bidirectional_LSTM)
* [**https://www.researchgate.net/publication/339645414\_Genre-ous\_The\_Movie\_Genre\_Detector**](https://www.researchgate.net/publication/339645414_Genre-ous_The_Movie_Genre_Detector)