**Movie Tags Prediction**

Predicting Tags for movies covers a wide range of information about movies. Being able to automatically generate or predict tags for movies can help recommendation engines improve retrieval of similar movies, and help viewers know what to expect from a movie in advance. Using Neural Networks model which merges the summary information and emotional flow through the summary to predict the sets of tags for movies. These tags can act as a strong search keywords and efficient features of recommended model. Tags for movies become convenient for movie recommendation to the end users depends on their preferences. This process would reduce the dependency of the human involvement to accumulate tags for movies.

This model uses the IMDb dataset which is the collection of movies and their respective synopsis. The reason behind selecting this particular dataset is two-fold. First, the tag set is comprised of manually curated tags. These tags express only plot-related attributes of movies (e.g. suspenseful, violence, and melodrama) and are free of any tags foreign to the plots, such as metadata. Furthermore, grouping semantically similar tags and representing them by generalized tags helped to reduce the noise created by redundancy in tag space. Second, the corpus provides adequate amount of texts in the plot synopses as all the synopses have at least ten sentences.

**Data pre-processing:**

Lowercase the synopses and remove the common words and limit the vocabulary to top 5k words to reduce noise. Then convert each synopses into 1500 integers where each integer represents the index of the corresponding word in vocabulary. For the synopses greater than 1500 words then truncate them whereas the shorter sequences are left padded with zeroes.

**Proposed Model:**

The proposed model consists of three modules which captures the emotional flow of the storyline and the text based representation of the synopsis to get the relevant tags for a movie.

The first module is a convolutional neural network (CNN) to learn the plot representation from synopsis. The second module works on the flow of emotions capture through Bi-directional long sort-term memory (Bi-LSTM) network. The final module is the hidden dense layers that operate on the combined representations generated by the previous modules to predict tags.

1. **First Module (CNN):**

This takes the word sequence as an input and each word is represented by 300 dimensional word embedding vector. We consider 4 sets of 1-d convolution models with 1024 filters and each filter of size 2, 3, 4 and 5. Filter of size ***C***from window ***t*** works on the window ***t-C+1*** on a word sequence x1, x2,…….xn. Filter of size C uses weight Wc and bias bc. After applying the activation functions and by applying the max over time pooling operation and takes the maximum value as the feature produced by a particular filter. The pooling operation outputs can be obtained by concatenating for four filter sets that represent the feature representations for each plot synopsis.

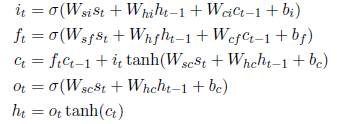
1. **Emotional capture:**

Here, we design a neural network architecture that tries to learn representations of plots using the vector space model of words combined with the emotional ups and downs of plots which helps to understand how the story unfolds. To model the flow of emotions throughout the plots, we divide each synopsis into N (N=20) equally-sized segments based on words. For each segment, we compute the percentage of words corresponding to each emotion and polarity type (positive and negative) using the NRC emotion lexicons. More precisely, for a synopsis x *E* X, where X denotes the entire collection of plot synopses, we create N sequences of emotion vectors using the NRC emotion lexicons as shown below:

x → s1:N = [s1, s2, ..., sN ]

where si is the emotion vector for segment i.

For Bi directional LSTM we send the emotional sequence data as an input. This bidirectional LSTM layer tries to summarize the contextual flow of emotions from both directions of the plots. The forward LSTMs read the sequence from s1 to sN, while the backward LSTMs read the sequence in reverse from sN to s1. These operations will compute the forward hidden states (h1,…….,hN) and backward hidden states (!h1,……….,!hN). For input sequence s, the hidden states ht are computed using the following intermediate calculations:



where, W and b denote the weight matrices and bias, respectively. i, f, o, and c are input gate, forget gate, output gate, and cell activation vectors, respectively. The annotation for each segment si is obtained by concatenating its forward hidden states hi and backward hidden states !hi , i.e. hi=[hi;!hi]. We then apply attention mechanism on this representation to get a unified representation of the emotion flow. Weighted sum r is computed on the top of feature map in attention layer.



Where 

and score is calculated as score(hi)=vTtanh(Wahi+ba)

where, W, b, v, and u are model parameters.

1. **Third Moule:**

In this module, we concatenate the output generated by CNN in the first module and the output generated by the attention layer in second module. They are fed into dense layers with 500 and 200 neurons. We are using the drop out rate 0.4 after each hidden layer. Finally we compute output layer with 71 neurons which gives the 71 tags. To overcome the imbalance of the

tags, we weight the posterior probabilities for each tag using different weight values. Weight value CWt.



Where D is the size of the training set and T is the number of classes. Mt is the number of movies having tag t in the training set. Normalize the output using softmax function.



Based on the ranking we select the top N tags (N=3/5/10)