Multi-Class Sentiment Analysis of Movie Reviews using Deep Learning

Pankti Joshi
Masters in Computer Science
Lakehead University
Thunder Bay, Canada
pjoshi@lakeheadu.ca

Abstract—**Sentiment Analysis is one of the significant applications of Natural Language Processing (NLP). Natural Language processing being a derivative of Artificial Neural Networks (ANN), can solve a lot of real-time problems such as sentiment analysis, speech recognition, or machine translation. Artificial Neural Networks with multiple hidden layers, which is known as Deep Neural Network (DNN), can solve a lot of predictive analysis problems in the field. The most functional topic in Deep Neural Network is Convolution Neural Network (CNN). It has its wide roots applications in Machine Learning, Computer vision, Deep Learning, Recommendation Systems as well as Natural Language Processing. This paper proposes a model to implement a Convolution Neural Network-based approach for text-based multi-class sentiment analysis for movie reviews. Various Natural Language Processing tools like Term Frequency-Inverse Document Frequency (TF-IDF), Bag of Words (BoW), Word2Vec, and Word Embedding are used to simplify text representation. There is a brief explanation regarding the hidden layers and comparative analysis of how do various parameters affect CNN efficiency.**

Index Terms—**Convolution Neural Network, Sentiment Analysis, Bag of Words**

I. Introduction

Movie reviews are a means to provide the performance of a movie. Nowadays, a lot of people rely on movie reviews and consider it a significant aspect of the success or failure of a film. A movie review can be of different kinds – it can be described in a single word, sentence, or phrase. A textual review would provide an in-depth analysis of the movie, its strengths, and weakness as well as comment on whether the film fulfills the viewer's expectations or not. CNN has achieved great results for dealing with extensive data, and its applications are enormous while dealing with texts based data. One of the research areas includes sentiment analysis in the field of NLP, where a lot of experimentation's are taking place.

Sentiment analysis is classifying the data based on emotional classification. It assigns positive or negative sentiments to a text. The Word2Vec and Word Embeddings are NLP techniques that are used to construct the linguistic contexts of the words, thus increases the overall accuracy of the model. Various NLP algorithms such as BoW, TF-IDF are used to simplify text representation and convert the text to the vector representation, which is then used to train the neural networks. CNN handles almost all the problems in NLP in an efficient manner.

Sentiment analysis is a significant research arena in the field of NLP targets to derive the actual emotions from the textual reviews[1]. It can be related to text mining, and the reviewer tries to categorize the polarity of the report as positive, negative, or neutral attitude. In the proposed model, an attempt has been made to use multi-class sentiment analysis to categorize the movie reviews given by the reviewers into respective categories using deep learning. The model has been constructed such that it is restricted to use only the CNN model with different layers of the neural network.

The dataset considered in the task is from Rotten Tomatoes Movie Review Dataset for multi-class classification of the movie reviews. The execution environment used is Google Colaboratory, and CNN is executed using Keras. The hyper parameters, as well as a number of input-output layers, have been changed and experimented with getting better accuracy results. Various loss functions have been used to calculate the loss of data while the model is trained and executed. The model's accuracy has been defined based on factors like accuracy, precision, and f1-score. A CNN model is proposed with different layers of neurons to get a multi-class sentiment analysis of the movie reviews.

Section II provides a short literature review of the work conducted in the research area related to the sentiment analysis using CNN. The details related to the dataset used in the proposed model are provided in Section III. Section IV explains the Proposed model and its architecture. A brief experimental result is provided in Section V, which is followed by Results and Conclusion, after which Appendix is mentioned, which provides explanations for the essential code snippets used in the proposed model.

II. Literature Review

In the research conducted by Charu Nanda, Mohit Dua, and Garima Nanda in the paper, "Sentiment Analysis of Movie Reviews in Hindi Language using Machine Learning."[3] The databases considered are the movie reviews that are in the Hindi language. The necessary steps mentioned in the research include data preprocessing, filtering, classification, and performance evaluation. Data Preprocessing steps mainly include the removal of stop words and special symbols that increases the complexity of the dataset. The preprocessed data is then seperated based on their polarity i.e., positive, negative, or neutral words. The classification algorithms such as Random Forest and Support Vector Machine (SVM) is used to classify the model. The performance of the model is evaluated based on factors like recall, accuracy, precision, and F-measure. The comparative analysis is shown between several approaches and several different algorithms, such as Naive Bayes, Resource-based, and Machine Learning. It has been concluded that the proposed model provides better accuracy with all the datasets, and SVM and Random Forest algorithms work well while doing the sentiment analysis of movie reviews, and provides an accuracy of 89.73 and 91 percent respectively.

In the research paper, ”Prediction of Movie Sentiment Based on Reviews and Score on Rotten Tomatoes Using SentiWordNet,” the authors Suhariyanto, Ari Firmanto, and Riyanarto Sarno proposes a new method to anticipate the sentiment of movie reviews on rotten tomatoes[4]. The technique would combine the score from SentiWordnet as well as the original expert score. It can be concluded that the proposed method could provide a better F score of 97 percent. The necessary steps mentioned include data preprocessing - remove punctuation, stop words removal, lemmatization, and tokenization. The next step is feature extraction using the Senti score from SentiWordNet and Original Expert Score. Sentiment Classification is done using Logistic Regression. The result is evaluated using recall, precision, and accuracy. The proposed method in the research is efficient as it could recognize the implicit values, whereas it is not good enough to handle the imbalanced dataset.

In the research, ”Enhancing the Performance of Sentiment Analysis by Using Different Feature Combinations” proposed by Nazma Iqbal, Afifa Mim Chowdhury, and Tanveer Ahsan states several varied ways to combine multiple features to provide a better performance of Sentiment Analysis[5]. In the paper, the author uses two different kinds of datasets, namely the Stanford Twitter Sentiment 140 dataset and the IMDb Movie Reviews dataset. Several machine learning algorithms were implemented, like SVM, MaxEnt, and NB. The results were evaluated based on accuracy, precision, recall, and F1-score. The steps proposed and executed include preprocessing the data and then feature extraction (consists of the BoW technique and word n-gram features). The feature combination is an additional measure included in the proposed model. These would try and combine techniques such as single word feature, stop word filtered word features, and bigram chi-square word features. Then feature selection occurs, after which machine learning algorithms are implemented on the preprocessed dataset. This would provide the sentiment classification model and put up the expected labels. To conclude, combining the feature technique gets better accuracy and f1-score by 2-5 percent as compared to the other models, in sentiment analysis tasks.

III. Dataset

The raw train dataset of Rotten Tomatoes movie reviews has been used to train and test the model. It is in the .tsv format and consists of several attributes like PhraseId, SentenceId, Phrase, Sentiment, and updated Review. The dimension of the dataset is 156060×4. Each sentences are broken down into phrases. In the dataset, the PhraseId and SentenceId identifies the phrases and sentences in the dataset. Phrase represents the whole textual phrase for reference. Each phase is associated with each sentiment based on the numerical number 0 to 4. Each numerical value in sentiment has its significance, 0 represents negative, 1 somewhat negative, 2 neutral, 3 somewhat positive, 4 as positive. The dataset has been broken down in a ratio of 70:30, for train and test data with a random state set to 2003. All the features have been considered to get the optimum accuracy from the model. This dataset is found in Github and initially collected by Pang and Lee. The table 1 below depicts the head five rows in the output obtained.

TABLE I
Dataset Head 5 rows

PhraseId	SentenceId	Phrase	Sentiment
1	1	A series of escapades demonstrating the adage that what is good for the goose is also good for the gander, some of which occasionally amuses but none of which amounts to much of a story.	1
2	1	A series of escapades demonstrating the adage that what is good for the goose	2
3	1	A series	2
4	1	A	2
5	1	series	2

IV. Proposed Model

The proposed model is a CNN based multi-class classifier for the movie reviews. The model has been implemented in Keras open-source neural network library, and the code is written in python. The model was initially executed in PyTorch, but due to space complexity issues, google colaboratory does not support the execution. Thus, Keras support highly complex data. Two major python libraries are used, pandas and NumPy. The Pandas library enables easy data manipulation and analysis. Numpy is used to support large mathematical matrics computations of data.

V. Proposed Model

There are several steps involved in building the proposed model. The primary step is to split the data into training and testing tuples in the ratio 70:30, with a random factor 2003. After that, preprocessing the dataset is done by doing stop words removal, lemmatization, stemming, and remove punctuations. After that, the CNN model is designed with different input layers like convolution layer, Dense layer, flatten layers, pooling layers like max-pooling, and average pooling and drop out layer to combat the issue of overfitting. Various optimizers were used and experimented like Adam,

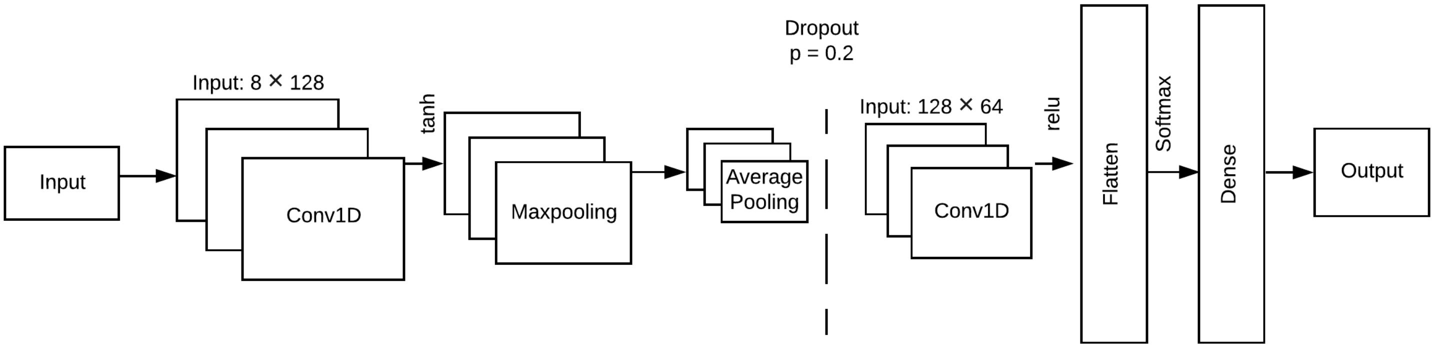

Fig. 1. Proposed CNN Architecture.

Adadelta, and adamax. Activation functions used include relu, softmax, and tanh. Secondly, as the input is textual data, the techniques such as word embedding and word2vec are applied to convert the words into a vector. Additionally, NLP algorithms such as BoW and TF-TDF plays a vital role while dealing with the movie reviews data. Lastly, the model is passed through the training phase, and we get the output in terms of accuracy, precision, recall, and f1-score. The following section V provides in detail explanation about the implementation of the proposed model. The base model is implemented using the TF-IDF method, and for achieving the optimum output results, changes have been made to the input layers in the CNN and the hyperparameters. The optimum results obtained from the model implemented is accuracy:62 , f1-score:61 , precision:68 , recall:50. Table 2 shows various configurations used in the proposed model. Figure 1 shows implementation of various input layers in the CNN network.

TABLE II
PARAMETERS CONFIGURATIONS OF PROPOSED MODEL

Parameter	Configurations
Optimizer	Adadelta
Activation function	ReLU, Tanh
Pooling Layers	MaxPooling;AveragePooling
Kernel	1
Batch size	512
Epoch	100
Softmax(dimension)	1
Dropout(Probability)	0.2

VI. IMPLEMENTATION

There are several crucial implementation steps in the proposed CNN model. Firstly, data preprocessing is a significant step in this process. This step includes cleaning the data by removing the null values, removing the stop words, and punctuations. All the sentences are converted to a lower case. The NLTK word tokenize library is used to tokenize the sentences.

Word Lemmatisation is done to get the actual root word from the words given. This is implemented using the WordNetLemmatizer package of the NLTK python library. The proposed model has been implemented using the Stemming process as well. But the accuracy of the model increases while executing the model using Lemmatization. This is because, in both the steps, the root words are obtained, but lemmatization provides a meaningful language word.

The data is then passed through the convolution layer. The convolution layer extracts the feature of the data provided, by taking as image matrix and filter or kernel as input. This data is then passed through an activation function, which would map the input to the response variable. In the proposed model, the tanh activation function has been applied to the extracted features. After this, the data is passed to the MaxPooling layer to extract the maximum value from each patch of the feature map. The optimized output is passed through the second Pooling Layer, AveragePooling, which would extract average values from each patch of the feature map. This data is passed through the dropout layer with a probability of 0.2. The dropout layer would reduce the overfitting problem caused. This output from the Dropout layer is fed into another 1D Convolution layer, after which additional Relu activation function is applied to the output data for a better score. The flatten layer would convert the pooled feature map to a single column to be passed to the fully connected layer. The output is passed through a softmax layer; this would allow the neural network to run through a multi-class function. This would enable a clear understanding of the identification of various sentiments in the textual data. These obtained features are then passed through the dense layer to achieve an output. The Adadelta optimizer is used to optimizee the output results. The Loss in the information is calculated using the categorical cross-entropy function; this would compute the Loss value in the output values of the proposed model. Accuracy, F1-score, Precision, and Recall values are used to calculate the efficiency of the model.

As the input data used in the proposed model is in the textual format, there is a need to convert the words in the vector format. The techniques, such as word embedding and word2vec, are applied to translate the words into a vector format. Additionally, NLP algorithms such as BoW and TF-TDF plays a vital role while dealing with the movie reviews data. The BoW model is used in NLP for natural representation and information retrieval. This would indicate the occurrence of the words in the document. Word embedding technique would map the phrases from the vocabulary to the vectors of real numbers. TF-IDF would indicate how a word is essential to a document in the collection of vocabulary. It is mainly used for information retrieval and text summarization.

For the proposed model, after passing through all the layers mentioned above, the accuracy and f1 score of the model is 61 and 60 percent respectively. The following section provides in detail explanation about the experimental analysis.

VII. Experimental Analysis

To achieve better outputs and to compute the predicted values nearer to the ideal values, various parameters such as number of epochs, learning rates, batch size, number of layers, optimizer used, ngram count, number of features considered, dimensions of the input and output can be altered. As mentioned in the previous sections, computing better results are done by hit and trial method- by setting different values for various parameters. Thus, the proposed model did go through several use cases to obtain the highest Accuracy. Multiple cases experimented are discussed in the section.

Primarily, the proposed model was executed without the Average Pooling layer and 200 epochs. All the other parameters were the same, as mentioned in the proposed model. The model gave Accuracy of 0.58 with Categorical entrophy loss as 0.96, while with the average pooling layer, it was 0.59 and 0.97, respectively. While changing the value of Dropout to 0.5 along with 100 epochs and without Average Pooling Layer Loss reduced to 0.89.

Secondly, the model has been tested using the CountVectorizer with accuracy of 59 percent. Also, there was a problem of over fitting while executing the model, this was overcome using the dropout layer and reducing the number of epochs.

A. Experimental Calculations

Though different kinds of variations in parameters, we can achieve different scores. This section provides a brief about the equations used and required to compute the TF-IDF formula. Equation 1 depicts the implementation of the TF-IDF to convert the words to the vectors.

$$tfidf_{i,d} = tf_{i,d} \cdot idf_i \tag{1}$$

The equation to compute the accuracy is provided in equation 2, given below. Accuracy is a measure to find the efficiency of the model.

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2}$$

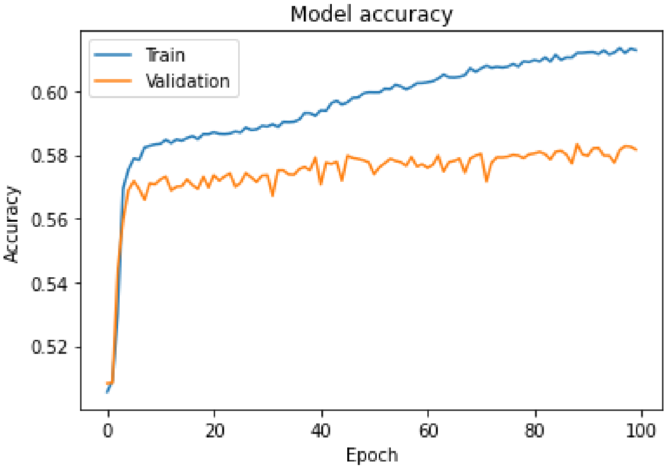

Fig. 2. Model accuracy in terms of Accuracy vs Epoch.

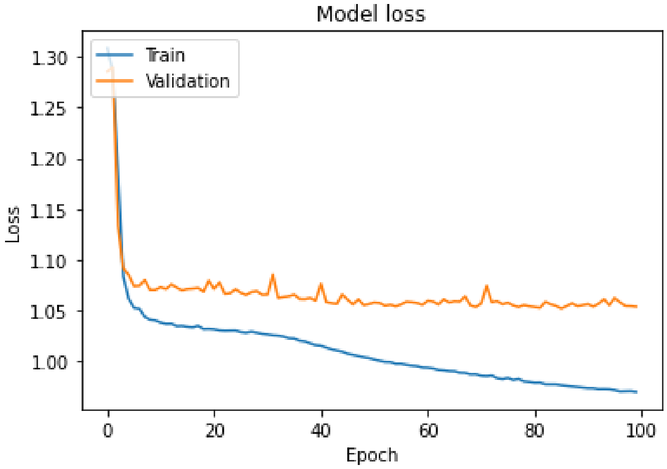

Fig. 3. Model Loss.

VIII. Experimental Analysis

To achieve better outputs and to compute the predicted values nearer to the ideal values, various parameters such as number of epochs, learning rates, batch size, number of layers, optimizer used, ngram values, no of features, dimensions of the input and output can be altered. As mentioned in the previous sections, computing better results are done by hit and trial method- by setting different values for various parameters. Thus, the proposed model did go through several use cases to obtain the highest Accuracy anf F1-Score. Multiple cases experimented are discussed in the section. The figure 2 shown below shows the accuracy of the proposed model in terms of the graphical representation inform of accuracy vs epoch.

IX. Result

The proposed model has been trained in a few way to get the optimum accuracy of the model. The hit and trial methods has been adopted to execute the mode. It can be concluded that the multi-class sentiment analysis can be executed using simple CNN model as well. The efficiency of the model is shown in the table below.

TABLE III
EFFICIENCY MEASURE

Parameter	Scale in Percentage
Accuracy	62
F1-Score	61
Precision	68
Recall	50
Categorical Entrophy Loss	96

X. CONCLUSION

A multi-class sentiment analysis model using deep learning has been implemented with an accuracy factor of about 60 percentage. A practical implementation of TF-IDF, BoW and CountVectorizer has been implemented. It can be concluded that the TF-IDF provides an optimum solution for these kind of classification.

XI. APPENDIX

A. Model Input Layers

```
cnnmodel = Sequential()
#cnnmodel.add(layers.Embedding(1715, 100))
cnnmodel.add(Conv1D(filters=128, kernelsize=1,
    activation='tanh',inputshape=(x_train_np.shape
    [1],x_train_np.shape[2])))
cnnmodel.add(MaxPooling1D(pool_size=2))
cnnmodel.add(AveragePooling1D(pool_size=2))
cnnmodel.add(Dropout(0.2))
cnnmodel.add(Conv1D(filters=64, kernel_size=1,
    activation='relu'))
cnnmodel.add(Flatten())
cnnmodel.add(layers.Dense(5, activation='softmax'))

cnnmodel.compile(optimizer="Adadelta",loss='
    categorical_crossentropy', metrics=['accuracy',
    f1_m,precision_m,recall_m])
cnnmodel.summary()
```

Listing 1. CnnRegressor Python class

REFERENCES

[1] Pang, Bo, and Lillian Lee. ”Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales.” In Proceedings of the 43rd annual meeting on association for computational linguistics, pp. 115-124. Association for Computational Linguistics, 2005.

[2] Firmanto Ari, and Riyanarto Sarno. ”Prediction of movie sentiment based on reviews and score on rotten tomatoes using SentiWordnet.” 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2016.

[3] Wankhede Rasika, and A. N. Thakare. ”Design approach for accuracy in movies reviews using sentiment analysis.” Sixth International Conference on Advanced Computing (ICoAe), 2014.

[4] Charu Nanda, Mohit Dua, and Garima Nanda in the paper, ”Sentiment Analysis of Movie Reviews in Hindi Language using Machine Learning.”13th International Joint Conference on Computer Science,2018

[5] Suhariyanto, Ari Firmanto, and Riyanarto Sarno, ”Prediction of Movie Sentiment Based on Reviews and Score on Rotten Tomatoes Using SentiWordNet,” International Journal of Engineering and Technology, 2018.

[6] Nazma Iqbal, Afifa Mim Chowdhury, and Tanveer Ahsan, ”Enhancing the Performance of Sentiment Analysis by Using Different Feature Combinations”,In 2016 3rd International Conference on Systems and Informatics (ICSAI), pp. 1062-1066. IEEE, 2016.