Multi-class Sentimental Analysis using Deep Learning

Akshaykumar Patel
Department of Computer Science
Lakehead University
Thunder Bay, Canada
apatel67@lakeheadu.ca

***Abstract*—This paper describes a multi-class sentimental analysis using a simple neural network architecture of five layers(Dense, Dropout, Flatten, Conv1D, MaxPooling1D).The main advantage of this model proposed is it justifies a problem of multi-class sentimental analysis based on Rotten Tomatoes movie review taking 70% of data for training and 30% of data for testing for the model.There are hyperparameters also defined, which also play an essential role in the outcome. Various experiments have been performed and got different outputs from the various experiments performed. These experiments have been performed by changing the layers of the model, also by changing the hyperparameters. The architecture of the CNN model is defined using the PyTorch libraries.The dataset used in for the above experiments is the Rotten tomatoes movie review dataset. The more the accuracy the better the model.After performing, nonlinear regression on different models, the model that is proposed gave an accuracy of 0.6318,loss of 0.9753,f1-score to be 0.6073,recall 0.5495 & precision of 0.6805this is the maximum i could find in this model**

***Index Terms*—ReLu,hyperparameters,epochs,dataset, normalization, dropout rate,convolution layer,batch size**

I. Introduction

In a world where we generate Quintilian bytes of data every day, sentimental analysis can be defined as feeding the algorithm with a massive amount of data so that it can adjust itself and continually improve. Sentimental analysis has become a key tool for making sense of that data. Automated process of understanding an opinion about a given subject from written or spoken language. Also known as Opinion Mining which is a field under Natural Language Processing(NLP) that builds system that try to identify and extract opinions within text(movie reviews).Now-a-days, sentimental analysis is a topic of great interest and development since it has many practical application. This is formally defined as the task to identify and analyze subjective information of people's opinion in social media sources(Pham and Le,2016:Yang et al., 2017).This field of study has recently attracted a lot of attention due to its implications for businesses and governments. The challenges of this task can be summarized in the following examples:

- For example, the scope of negation, the polarity of an adjective cannot be determined. Here there could be long-distance dependencies on the context and the domain.
- There could not be explicit way of revealing their opinion as they could be indirect, subtle or ironical.
- For example, a "short cord" mostly indicates a negative opinion, while a "short boot time" signifies a positive one.

Users generally express a broad variety of sentimental with a wide range of degrees, but to simplify the task, sentimental analysis approaches have traditionally classified sentiments into positive, negative or neutral. This paper describes system for multi-class sentimental analysis using Deep Learning. I evaluate my system on a dataset using the web-URL. For defining in a way, I extracted first phase and sentiment & then text and sentiment. On the part of pre-processing steps such as Removing Stop words as well as punctuation marks, converting string into lower case, Lemmatization, converting string into array using term frequency-inverse document frequency(TF-IDP) which I observed to be the most efficient way of converting. Finally, I used keras to build the model with five layers. Our source-code publicly available at https://github.com/apatel67/Assignment-2-Keras-Multi-class-Sentimental-Analysis.git

II. LITERATURE REVIEW

A method to analyze these reviews is the text sentiment analysis (also named opinion mining). Sentiment analysis requires machine learning related knowledge and Natural Language Processing technique. Many machine learning algorithms were researched and developed to apply in the sentiment analysis task. Sentiment analysis has been an important subtopic of natural language processing (NLP). There are many specific small research directions in text sentiment analysis. One main sub research direction is the text sentiment classification. In text sentiment classification, many different methods have been researched to solve the task.Sentiment analysis (also known as opinion mining or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affected states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.On the part of sentimental analysis,a classification problem uses evaluation metrics of Precision, Recall,F-score & accuracy.Depending on the balance of classes of the dataset the

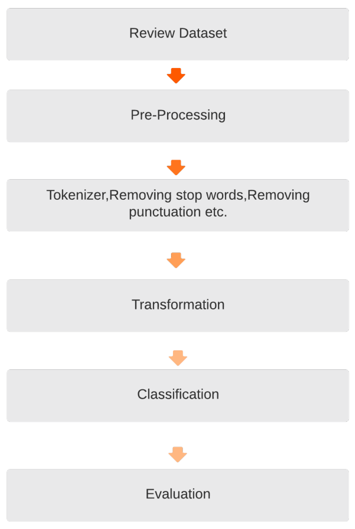

Fig. 1. Steps to evaluate Sentimental Analysis

most appropriate metric should be used.Flowchart Fig.1 given below states the steps to evaluate sentimental analysis[2].

III. RELATED WORK

Sentiment analysis refers to the process of extracting explicit or implicit polarity of opinions expressed in textual data(e.g., social media including online consumer reviews [1, 7]).Existing approaches to sentiment analysis can be grouped into three main categories: knowledge-based techniques, statistical methods, and hybrid approaches.Knowledge-based techniques classify text by affect categories based on the presence of unambiguous affect words such as happy, sad, afraid, and bored. Some knowledge bases not only list obvious affect words, but also assign arbitrary words a probable ”affinity” to particular emotions. Statistical methods leverage elements from machine learning such as latent semantic analysis, support vector machines, ”bag of words”, ”Pointwise Mutual Information” for Semantic Orientation.Traditional sentiment analysis focus on extracting opinion polarities at a coarse level, which cannot fully satisfy aforementioned purposes. Sentiments are normally domain dependent (e.g. delicious indicates positive sentiment in the food domain, where it does not indicate any sentiment in the laptop domain). Additionally, consumers tend to express their sentiment regarding/associated to specific features or aspects of different goods or services. Extracting sentiments with consideration of the associated aspects is termed as Aspect Based Sentiment Analysis (ABSA)[3].Specific study on the task of multi-class classification of online posts of Twitter users, and show how far it is possible to go with the classification, and the limitations and difficulties of this task[4].Sentiment analysis and opinion mining in social networks present nowadays a hot topic of research.A novel paper on this is classification of texts collected from Twitter and classifies these texts into multiple sentiment classes[5].However, the aforementioned transfer learning models do not support multi-label classification natively[6]. It is because the softmax function used in the output layers of the models only support single-label classification tasks. Essentially, the softmax function produces a probability distribution over all the classes, where only one class with the highest probability will be selected as the output. This limitation must be addressed since multi-label classification tasks require the predictions to have more than one classes.

IV. DATASET

The dataset which i am using for multi-class sentimental analysis is master dataset of sentimental analysis on rotten tomatoes movie review with columns as phrase & its ID , sentiment, sentence & ID.The main aim was to justify movie reviews which were text based about their accuracy of the model generated. I have performed pre-processing by first tokenize the sentence, remove the stop words and punctuation marks, converting string into array,Lemmatization etc.Further, the data has been split into training and testing; it is done by using the train test split package of the sklearn.model selection library. It is split into 70 percent and 30 percent, i.e., 70 percent data in the training data segment and 30 percent data in the testing data segment.Now, both the data (i.e., training and testing) are converted into a NumPy array using training and testing, the benefit of converting it into a NumPy array is that the data then becomes easy for manipulation. Table. 1 and Table. 2 shows the comparison between older data which was in data frame and generation of new index label each with phrase and sentiment respectively.

TABLE I
RANDOM INDEX LABEL GENERATION

	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage...	1
1	2	1	A series of escapades demonstrating the adage...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2

TABLE II
NEW LABELS GENERATED

	PhraseId	SentenceId	Phrase	Sentiment
0	116057	6186	please not only the fanatical adherents on eit..	1
1	123040	6603	to overwhelm everything else	1
2	17176	746	The large-frame IMAX camera lends itself beaut...	4
3	48914	2388	between the film’s creepy	2
4	67498	3427	sends you out	2

V. PROPOSED MODEL

The model which i have proposed comprises of many different layers while doing this experiment.The first two layer is Conv1d layer.This layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs[8].It applies a 1D convolution over an input signal.The next one is MaxPooling1d layer.In this layer we have to define the pool size & strides specifically as this operation that calculates the maximum, or largest, value in each patch of each feature map.The primary purpose of the convolution and pooling layers is feature extraction.Moving on to next layer Dropout which provides a function of randomly selected neurons are ignored during training.Basically it is regularization technique for neural network models proposed by Srivastava, et al. in their 2014 paper[9] which is a simple way to prevent neural network from over fitting.Rate defined is for setting good value for dropout in hidden layer here in this case it is 0.5.Next layer is flatten layer which collapses the spatial dimensions of the input into the channel dimension for sequence input only.Last layer is dense layer.A dense layer is just a regular layer of neurons in a neural network. Each neuron receives input from all the neurons in the previous layer, thus densely connected.For Naive Bayes classifier-based sentimental analysis comes into the picture when dealing with the theory.Here flow chart for each layer used in the proposed model is shown in Fig.4

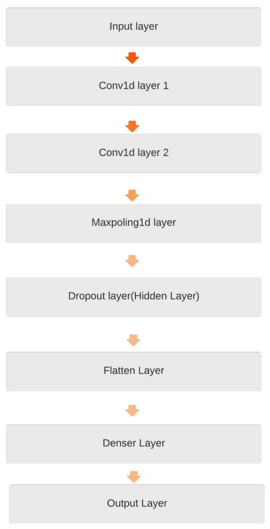

Fig. 2. Model Summary

VI. EXPERIMENTAL ANALYSIS

While working with the code, several different combinations of methods were applied to it. These methods mean changing different parameters in the code, which include: batch size,number of input layers, number of classes, parameters of input layers,random state ,dropout rate, and more. The parameters mentioned above are also known as hyperparameters. I experimented by changing a lot of different hyperparameters, like, epochs,parameters of the input layer, batch size, dropout rate, and more. The most impact my model had was when I changed , batch size, epochs, and optimizer.Whole process of model works like first data framing,pre-processing steps such as removing punctuation marks, removing stop words, using lancaster stemming & Lemmatization.Stemming and Lemmatization are Text Normalization (or sometimes called Word Normalization) techniques in the field of Natural Language Processing that are used to prepare text, words, and documents for further processing.Below text shows how it converts word in data frame to words after pre-processing,for ex. [[”’s”, ”stupid], 1] = [[”$'sstupid$”], 1].

VII. CONCLUSION

In this paper,model was develop with many different layers for the problem of multi-class sentimental analysis using two activation function Relu(Rectified Linear Unit) and Softmax.Sentimental analysis is reference to the task of Natural Language Processing to determine whether a text contains subjective information and what information it expresses i.e., whether the attitude behind the text is positive, negative or neutral.After meticulous experimental analysis, by taking batch-size as 64 number of classes as 10 and with 20 epochs the model has been found out:

Accuracy : 0.6318
Precision : 0.6805
Recall : 0.5495
F1-score : 0.6073

REFERENCES

[1] Fang X, Zhan J. Sentiment analysis using product review data. J Big Data. 2015;2(1):5.
[2] Symeon Symeonidis, Democritus University of Thrace 5 Things You Need to Know about Sentiment Analysis and Classification.
[3] Pham DH, Le AC. Learning multiple layers of knowledge representation for aspect based sentiment analysis. Data Knowl Eng. 2018;114(January 2017):26–39.
[4] M. Bouazizi and T. Ohtsuki, ”Multi-class sentiment analysis on twitter: Classification performance and challenges,” in Big Data Mining and Analytics, vol. 2, no. 3, pp. 181-194, September 2019.
[5] M. Bouazizi and T. Ohtsuki, ”A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter,” in IEEE Access, vol. 5, pp. 20617-20639, 2017.
[6] Tao, J., Fang, X. Toward multi-label sentiment analysis: a transfer learning based approach. J Big Data 7, 1 (2020).
[7] Akhtar MS, Gupta D, Ekbal A, Bhattacharyya P. Feature selection and ensemble construction: a two-step method for aspect based sentiment analysis. Knowl Based Syst. 2017;125:116–35.
[8] https://keras.io/layers/convolutional/
[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov Dropout: A Simple Way to Prevent Neural Networks from Overfitting:15(56):19291958, 2014.

VIII. APPENDIX

```python

for l in range(len(documents)):
  reviewlabel = documents[l][1]
  newReview = []
  for w in documents[l][0]:
    newWord = w
    if removePuncs and (w in punctuations):
      continue
    if remove_stopwords and (w in stopwords_en):
      continue
    if useStemming:

      newWord = lancaster.stem(newWord)
    if useLemma:
      newWord = wordnet_lemmatizer.lemmatize(newWord
    )
    newReview.append(newWord)
  documents[l] = (newReview, reviewlabel)
  documents[l] = (' '.join(newReview), reviewlabel)

print(documents[0])
```

Listing 1. Pre-Processing steps