MULTI-CLASS SEMANTIC ANALYSIS USING DEEP LEARNING

Krishna Sanjaykumar Patel
Student ID: 1111405
kpatel63@lakeheadu.ca
Dept. Masters of Computer Science
Lakehead University [0]

Abstract— **An important aspect of our knowledge gathering behaviour has always been to figure out what other individuals think. With the increasing availability and prominence of opinion-rich resources, such as online review and social networking sites, new possibilities and challenges emerge as individuals can, and should, actively use information technology to seek out and understand the opinions of others. The aim of sentiment analysis is to utilize automated tools to recognize subjective material such as views, attitudes and feelings expressed in the form of text.**

This research paper focuses on implementation of an one-dimension convolution (Conv1D) based neural network for a systematic sentiment evaluation of the text-based movie critics. The model performs multi-class sentiment analysis over the rotten tomatoes movie review corpus (raw data-set). The model is programmed to measure its performance using accuracy, precision, recall and F1 Score by training and testing the data.

Keywords— **Sentiment analysis, Naive Bayes, Support Vector Machine, Movie Reviews, Natural Language Processing, Machine Learning, Deep Learning, Neural Network.**

I. INTRODUCTION

In recent years, the usage of various machine learning methods for the resolution of Natural Language Processing (NLP) issues has become a significant phenomenon. One of these problems is automated recognition (positive, negative, neutral) of emotional colouring in the text data, i.e. the sentiment analysis. This task is intended to decide whether a text (movie review) is positive, negative or neutral, based on its impact on the credibility of a particular film.

Performance and efficiency challenges are a driving force in the growing interest of businesses and analysts towards improving sentiment analysis. The sentiment analysis tends to be one of the most demanded tasks for the NLP. For example, various international contests and competitions [1] attempt to find the best strategy of identifying emotions. Sentiment analysis has been carried out on various levels, beginning from the whole text level and continuing to the sentence level or phrase level.

II. LITERATURE REVIEW

The study of sentiments analysis has very little history. Reference [2] is widely acknowledged as the primary research on using text classification machine learning techniques for sentiment analysis. The earlier studies in this area involves methods focused on maximum relative entropy, binary linear classification [3] and unsupervised learning [4].

Most of these methods use popular features such as bag-of-words, n-grams and tf-idf which are believed as the straight-forward one. However, as the findings of the experiment demonstrate, simple models still perform better than complicated ones. Reference [5] demonstrates obtain symbolic data by using remote information. In comparison, as they primarily interact for movie comments and tweets, they used extra features such as positive emoticons ":)"":-)" as positive and negative emoticons such as ":("":-(" as negative. They program algorithms using Naive Bayes, MaxEnt and Support Vector Machines (SVM) and evaluate SVM over other classifiers.

The other attribute is syntactic meta-information. It is clear that the iteratively enumerable grammar repre-

sents the most comprehensive of all natural languages. The statistical efficiency of the best lexical context free grammar parser is linear, so syntactic knowledge is costly for performing sentiment analysis. However, the studies concerning dependence relationships have shown that syntax leads greatly to both Recall and Accuracy of most algorithms.

III. PROPOSED CNN MODEL

A. Data-set

The Rotten Tomatoes movie review corpus is a collection of film reviews accumulated by Pang and Lee in [6]. The corpus consists of around 150,000 sentences. This corpus was evaluated in [7] as the tree structure parses each sentence and a fine-grained emotion label of 0 to 4 with the amount being very negative, negative, neutral and positive, or very positive as shown in figure-1. In this paper we use this data in a number of algorithms to teach a program to interpret emotions accurately.

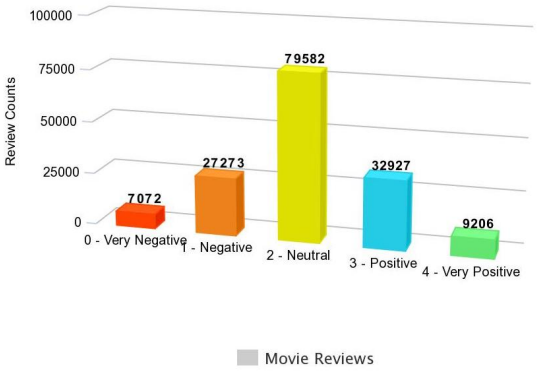

Fig. 1: phrase and sub-phrase bar graph

All the approaches described in this paper are evaluated by training on a random sub-set of phrases (and sub-phrases) approximately 7/10 in size of the data set and testing on the remaining 3/10. The most popular algorithms for fine-grained sentiment analysis are able to perform with accuracy of 80.7 percent (see [7]).

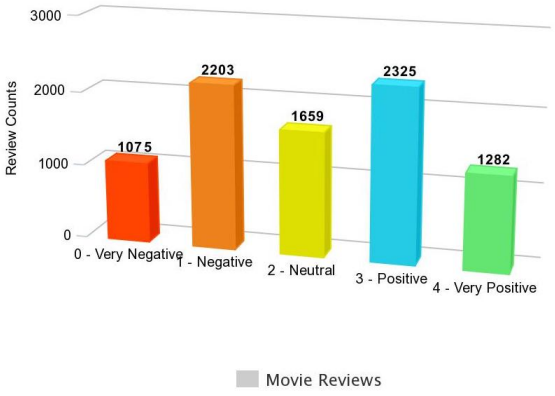

Fig. 2: whole phrase bar graph

The Rotten Tomatoes movie review corpus has phrases and sentences as shown in table-1. Each phrase has a phrase ID and each sentence has a sentence ID. Phrases that are repeated (such as short/common words) are only included once in the data. The number of actual sentences count with their sentiment are represented using a bar graph in figure-2.

Table 1: Details of Movie Review Corpus

Data Columns	Details
PhraseID	156060 not-null int64
SentenceID	156060 not-null int64
Phrase	156060 not-null object
Sentiment	156060 not-null int64
RangeIndex	156060 entries

B. Libraries

In order to perform multi-class Semantic Analysis using 1D-CNN, use of many pre-defined libraries have been done. Pandas library is used for data-frame and series data manipulation. As some mathematical calculation are required to perform over data-set in order to convert the pandas data-frame into array Numpy library has been used. SkLearn Library has been used to divide the data-set into train data and test data.CountVectorizer has been imported from SkLearn library as it helps to convert collection of text docs to a matrix of token

counts. nltk library helps to perform certain data pre-processing steps such as tokenization, removing stop-words, stemming, lemmatization and removing punctuation. Below are the list of libraries used while developing the CNN model. Keras is used to develop a 1D-CNN model.

- TensorFlow
- Pandas
- SkLearn
- Numpy
- nltk
- Keras

C. Data Pre-processing

Pre-processing data is a proven way of tackling various sort of problems. Pre-processing data prepares original data for further processing. Data pre-processing is used for database-driven systems such as client experience management and directive-based applications (such as neural networks).

- Tokenization:
  Tokenization is the method of dividing a text stream into words, phrases, icons, or other symbolic items called tokens. These tokens are very useful for finding such patterns as well as considered as a base step for stemming and lemmatization.
- Stop Words Removal:
  Text can include stop words such as' the," is," are.' Stop words should be extracted out of the text to be processed. There is no standard set for stop words in Natural Language Processing (NLP) research, but the nltk module includes a list of stop words.
- Punctuation Removal:
  Punctuation includes period, comma, exclamation point, question mark, colon, semicolon, bullet point, dash, hyphen, brackets, circle, shield, ellipse, quotation mark, and apostrophe. It is very important to remove punctuation before processing the data for further analysis. We have used nltk toolkit to remove punctuation from the data corpus.
- Stemming:
  Stemming is a method of reducing a word to its word stem that adds to suffixes and prefixes or to the roots of words such as lemma. Stemming is essential for natural language understanding (NLU) and natural language processing (NLP).
- Vectorization:
  Word embedding or word vectorization is one of the most common text vocabulary representations. It is capable of catching the meaning of a term in a document, the textual and syntactic similarity, the association with other terms, etc. We have used TF-IDF Vectorizer. TF-IDF is an acronym for Term Frequency-Inverse Document Frequency and is a very simple algorithm that transforms text into a realistic representation of numbers. The technique is commonly used to derive features across different NLP applications. The equation to calculate TF-IDF is shown in equation-1

$$W_{i,j} = TF_{i,j} \times \log \frac{N}{df_i} \tag{1}$$

Where,

$W_{i,j}$ = TF-IDF weight of i in j.
$TF_{i,j}$= Total number of occurrence of i in j.
df_i = Total number of documents containing i.
N = Total number of documents.

D. 1D-CNN Model

The creation of a convolutional neural network (CNN) is a perfect way to use deep learning for prediction. The Keras Library in Python allows it pretty simple to create CNN. At first, the raw data is converted to NumPy array using NumPy library. Various CNN Layers have been used such as Convolution layer, MaxPooling layer, Flatten layer etc. Given table-2 shows the attributes of respective layers.

Optimizers adjust the weight variables to decrease the error function. Loss feature serves as a reference to the field that informs the optimizer whether it goes in the right direction to hit the bottom of the range, the global minimum.
Keras have many optimizer such as AdaDelta, Adagrad,

Table 2: CNN Layers Attributes

Convolution Layer	
Filters	64
Kernel Size	3
Activation	relu
Input Shape	(2000, 1)
Convolution Layer	
Filters	128
Kernel Size	5
Activation	relu
MaxPooling Layer	
Pool Size	1
Flatten Layer	

SGD, Adam, Adamax etc. We have tested our model using various of optimizers and best results are taken into consideration along-with the best learning rate. Batch size of 64 is used to define our model.

E. Saving the model

It is very important to save a model. As it helps to avoid the unnecessary training time. You can resume your model from where you left off.

IV. Experimental Analysis

We compared our model by adjusting different analytical parameters such as batch size, optimizer, learning rate, Epoch, number of layers etc as shown in table-3. Changes in batch size will defer the results to some extent. What optimizer is used may also play a significant role in making the model more efficient. You may also make changes with the parameter of same optimizer to get a different result. Learning rate improvements are often considered to render our model more competitive.

Optimizers are algorithms or methods used to transform 56+roper batch size outcomes to the maximum r2 value and the lowest L1 loss. I applied the batch size to the optimizers and then measured the performance. When I allocated batch size 64, epoch to 25, and the optimizer was Adadelta, I had the maximum accuracy of 61.8 percent. The relation of optimizers with respect to batch size is seen in table-4.

Table 3: Optimum Analytical Parameters

Parameter	Value
Optimizer	Adadelta
Epoch	15
Batch Size	128
Number of Layer	2
Loss Function	categorical_crossentropy

Table 4: Comparison of Optimizer and Batch Size and Accuracy

Batch Size	Optimizer	Inference Time	Accuracy
264	Adadelta	494.72	0.608
128	Adadelta	503.39	0.613
64	Adadelta	483.17	0.618
264	Adam	471.89	0.601
128	Adam	494.52	0.612
64	Adam	483.03	0.617

The accuracy and L1 loss often depends on the amount of epochs allocated. When I carried out an observational study, I came to the conclusion that precision decreases as the magnitude of the epochs rises. At the other hand, the value of L1loss should decline as the values of the epochs are raised.

V. Applications

CNN models are used to provide the best outcomes in procedural decoding, search query extraction, inference, sorting, conventional NLP functions, and so on. Often, 1-D convolutions are used on time series in the frequency domain using an unsupervised algorithm to find anomalies throughout the time domain.
Classification and prediction are both two methods of data processing that can be used to derive models representing essential data groups or to anticipate potential data patterns. Classification is a data mining (machine learning) methodology used to determine community composition for data instances. Additionally, classification is often used for speech recognition, handwriting

recognition, bio-metric identification, paper classification and so on.

When addressing opinion analysis techniques, it is a powerful resource that can be utilised in social networking, user reviews and company customer support. It helps to assess public sentiment on an incident or object. This is often used to monitor social media and vocabulary input from users, poll responses, competitors, etc.

VI. Conclusion

The goal of this paper was to recognise optimizations and other strategies to classifying the sentiment of film reviews. Approaches are based on attempting to increase the quality of the test results.

In this paper, a model is developed demonstrating practical implementation of a 1D-CNN has been performed. Model is specifically developed to forecast the sentiments of the Rotten Tomatoes movie review corpus. Various experiments have been carried out by making various changes with certain parameters in the model. As a result of experimental analysis batch size of 64 and 2 convolution layers have been used providing the optimum results. Relu and SoftMax activation function are taken into consideration while developing the CNN model in order to achieve the maximum accuracy. As a result, the model provides the results with a precision of 65.32 percent and 1.011, 0.6183, 0.5894, 0.5362 values for loss factor, accuracy, F1 measure and recall respectively.

References

1. I. Chetviorkin, P. Braslavskiy, N. Loukachevich, "Sentiment Analysis Track at ROMIP 2011", Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialog 2012", pp. 1-14, 2012.
2. P. Bo, L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts", Proceedings of the ACL, 2004.
3. T. Joachims, "Text categorization with support vector machines: Learning with many relevant features", European Conference on Machine Learning (ECML) Springer, pp. 137-142, 1998.
4. P.D. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews", Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), pp. 417-424, 2002.
5. A. Go, R. Bhayani, L. Huang, "Twitter Sentiment Classification Using Distant Supervision", Technical report Stanford, 2009.
6. Bo Pang , Lillian Lee. Seeing stars: exploiting class relationships for sen- timent categorization with respect to rating scales, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, p.115-124, June 25-30, 2005, Ann Arbor, Michigan
7. Socher, Richard, Perelygin, Alex, Wu, Jean Y., Chuang, Jason, Manning, Christopher D., Ng, Andrew Y., and Potts, Christopher. Recursive deep models for semantic com- positionality over a sentiment treebank. In Con- ference on Empirical Methods in Natural Language Processing , 2013b.