DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5388

Volume 9, No. 1, January-February 2018

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

HYBRID ARCHITECTURE FOR SENTIMENT ANALYSIS USING DEEP LEARNING

Arnav Chakravarthy
Department of Computer Engineering
SVKM NMIMS Mukesh Patel
School of Technology Management and Engineering,

Simran Deshmukh
Department of Computer Engineering
SVKM NMIMS Mukesh Patel
School of Technology Management and Engineering,

Prssanna Desai
Department of Computer Engineering
SVKM NMIMS Mukesh Patel
School of Technology Management and Engineering,

Surbhi Gawande, Prof. Ishani Saha
Department of Computer Engineering
SVKM NMIMS Mukesh Patel
School of Technology Management and Engineering,

Abstract: Sentiment analysis involves classifying text into positive, negative and neutral classes according to the emotions expressed in the text. Extensive study has been carried out in performing sentiment analysis using the traditional 'bag of words' approach which involves feature selection, where the input is given to classifiers such as Naive Bayes and SVMs. A relatively new approach to sentiment analysis involves using a deep learning model. In this approach, a recently discovered technique called word embedding is used, following which the input is fed into a deep neural network architecture. As sentiment analysis using deep learning is a relatively unexplored domain, we plan to perform in-depth analysis into this field and implement a state of the art model which will achieve optimal accuracy. The proposed methodology will use a hybrid architecture, which consists of CNNs (Convolutional Neural Networks) and RNNs (Recurrent Neural Networks), to implement the deep learning model on the SAR14 and Stanford Sentiment Treebank data sets.

Keywords: Sentiment Analysis, Deep Learning, CNN, RNN, Word Embedding, Word2Vec

I. INTRODUCTION

In today's world, reviews of a service provided by a business play an integral role in determining its success. From a restaurant on Zomato to a product on Amazon, reviews are invariably used by customers while making decisions about which products or services to purchase. These days social media also plays a large role in providing a platform for people to express their opinions regarding businesses. Hence a technique that can classify these opinions and emotions into distinct classes or categories would be invaluable to any business, as they would no longer have to pore over millions of reviews manually to understand the success of their service as per the voice of their customers. Sentiment analysis is such a technique, in which text is broadly classified into three classes; positive, negative and neutral. Extensive research has been done on sentiment analysis using a feature selection approach. The main focus of this paper is to discuss a proposed methodology for implementing sentiment analysis using deep learning techniques, which is a relatively unexplored domain. We will be using the Stanford Sentiment Treebank data set and the SAR14 movie review dataset to test our model and optimize accuracy. The Stanford Sentiment Treebank dataset consists of 11855 sentences extracted from Rotten Tomatoes movie reviews and has fully labeled parse trees with five classes as labels (extremely negative to extremely positive). It has 8544 training examples, 1101 validation samples and 2210 test cases. The SAR14 data set is a data set of 233600 movie reviews

The input data will first be pre-processed, followed which a technique called word embedding will be performed. The resulting output will then be fed into a deep neural network architecture. In order to achieve optimal accuracy, we propose a hybrid architecture made up of CNNs and RNNs.

II. RELATED WORK

Alper Kursat Uysal et al. in their paper-**Sentiment classification: Feature selection based approaches versus deep learning** discusses a comparative study on two types of approaches: feature selection based and deep learning based. Experiments were conducted using four datasets with varying characteristics. For three out of the four datasets, various deep learning models outperformed feature selection based approaches.[1] Another paper- **Recurrent Neural Networks for Sentiment Analysis** written by Joaquın Ruales et al. compares LSTM Recurrent Neural Networks to other algorithms for classifying the sentiment of movie reviews obtained from the Large Movie dataset. It then repurposes RNN training to learning sentiment-aware word vectors useful for word retrieval and visualization.[2] Arman S. Zharmagambetov et al. in their paper- **Sentiment analysis of a document using deep learning approach and decision trees** talk about Google's algorithm Word2Vec, the various preprocessing steps required in the approaches, Deep RNN and the comparison between Deep Learning Approach and Decision Trees Approach and concludes that despite the fact that deep learning approach is more complicated rather than simple methods as "bag-of-words", it shows slightly better results.[3]

Feature level Sentiment Analysis on Movie Reviews written by Pallavi Sharma et al. talks about classifying the polarity of the movie reviews on the basis of features by handling negation, intensifier, conjunction and synonyms with appropriate pre-processing steps. The work proposed in this paper has given an accuracy of 81%, the performance is decreased because of not handling the True Negative reviews correctly.[4]

Houshmand Shirani-Mehr in his paper- **Applications of Deep Learning to Sentiment Analysis of Movie Reviews** compares the performances of different deep learning techniques as well as the implementation of the traditional Naïve Bayes classifier on performing sentiment analysis of movie reviews obtained from the Stanford Sentiment Treebank. [5]

III. TRADITIONAL APPROACH

The traditional sentiment analysis approach uses a bag-of-words approach to represent sentences and then feeds the fixed length vectors obtained as a result of this to a classifier such as Naive Bayes and SVMs. However, this approach fails to take into consideration the order of the words and hence leads to a less accurate model. Another modelling technique used is n-grams, but this has a fixed range and cannot be used for long term dependencies. [6] These approaches also often fail to classify sentences with a negation, such as 'The movie was not amazing' as negative and often give it a positive label.

A deep learning approach can overcome these problems. Word embedding provides contextual information about words and hence allows us to gain intuition about the sentences. Deep neural networks such as RNNs allow us to process sequential data, so information about the order of the words is retained and CNNs allow us to include phrase information in a word by word representation.

IV. PROPOSED METHODOLOGY

A. *Word Embedding*

A neural network cannot efficiently process text input as it cannot perform operations such as multiplication and convolution on words. Hence, we use a technique called word embedding, where the text is represented as vectors in space such that the distance between vectors depends on the semantic similarity between the words. Hence word embedding produces an embedding matrix of the input words which captures their meaning, semantic and contextual relations. To perform word embedding on our input data we will use the word2vec model. Word2vec is a two-layer shallow neural network that takes a text data set as input and first constructs a vocabulary from the text data. [3] It then learns the vector representation of these words and produces the word vectors as output. The output generated by the word2vec model will be the input to the neural network architecture.

B. *Hybrid Architecture*

This is a novel approach which applies CNN model followed by an RNN model. The model consists of an initial convolutional layer, a middle recurrent network layer and a final fully connected layer. This allows us to include phrase information in a word by word representation, obtained by CNN, which act as time steps for the RNN. We will test this by using a traditional RNN as well as an LSTM, to see whether the vanishing gradient problem makes a difference in our model and improves the accuracy.

1) CNN:

Convolutional Neural Networks (CNN) are powerful deep models for understanding image content, and have recently started being used in numerous classification problems[7]. CNN's have also started being used in text classification problems given their great performance.

Word2vec, proposed by Google, is a two-layer neural network model, which transforms words into vectors rather than sampling each value from a uniform distribution. These vectors are arranged in the vector space such that words with similar These vectors are fed to the CNN as input.

i) Convolution operation

Features obtained from N-grams of varying length play different roles in the final decision of the sentiment classification. Let's take a sentence "An adaption of *Ministry of Utmost Happiness* would probably have been better as a movie than a TV show", as an example to explain this better. In this case 3- grams would probably detect "have been better" as a positive indicator, whereas 5-grams would mostly detect "would probably have been better" as a negative indicator.

A convolution operation involves a filter f which takes a window of h word embeddings to produce a new feature c_i [8].

$$\chi_{1:n} = \chi_1 \oplus \chi_2 \oplus \ldots\ldots \oplus \chi_n \text{ [8]}$$

$$c_i = f(w \cdot \chi_{i:i+h-1} + b) \text{[8]}$$

b is the bias term. f is a non-linear function such as a hyperbolic tangent. A feature map *c* is constructed by applying this filter to each possible window of words in the sentence.

$$c = [c_1, c_2, \ldots. c_{n-h+1}] \text{ [8]}$$

ii) Max Pooling Operation

The feature maps obtained are now passed over to a max over time pooling operation layer. This step is used to select the most important feature, the one with the highest value. This pooling operation is applied to all the feature maps to obtain a vector with the most important features

$$c_{max} = max\{C\} = max\{c_1, c_2, \ldots. c_{n-h+1}\} \text{ [8]}$$

By using this scheme, we naturally deal with variable sentence lengths.

iii) Dropout and Softmax

To prevent overfitting of parameters and assigning too much weight to a particular node, we use dropout. Dropout randomly drops out parameters of hidden units in the classifier. This sets a portion of the features pooled in the previous layer to zero, so that only the unaffected units play a role in calculating gradients when passed to the softmax layer[8].

The softmax layer uses the features which were regularized using dropout, and computes the probability distribution of the input over all the labels, in our case the sentiments. This layer squashes a K dimensional vector contain real values into a vector containing values in the range [0,1], that add up to 1.

2) RNN:

Recurrent Neural Networks(RNNs) are powerful deep neural networks. Unlike traditional feedforward neural networks, RNNs have memory to a limited capacity, that is, the current input at a particular time step depends on the previous inputs as well. This makes RNNs an ideal tool for sequential data or data in which the order matters. Sentiment analysis is such an application, as the order in which words appear often change their contextual meaning. For example, in the sentence “The movie was not good” the word ‘not’ appearing before ‘good’ gives the sentence a negative sentiment.

RNNs have loops in them which allows information to be carried across the network as input is processed’[9]. An RNN can be depicted by figure --

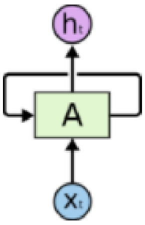

Figure 1. Rolled-up RNN [10][9]

To make it easier to understand, an RNN can be ‘unrolled’ into a series of connected feedforward networks and depicted as shown in figure --.

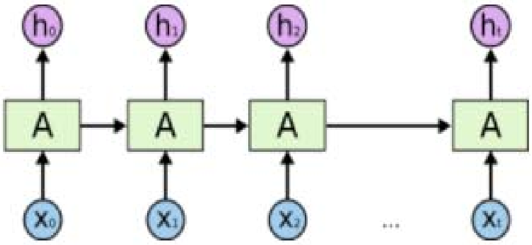

Figure 2. RNN Unrolled [10]

X_t refers to the current input at time step t, which in our case is the t^{th} word of a given sentence represented as a one hot vector. A is the hidden layer, which is recurrent across the network, and h_t refers to the output at time step t, which is a function of the current input as well as the output of the previous input. h_t is calculated as,

$$h_t = \sigma\, (W^H h_{t-1} + W^X x_t) \quad [11]$$

where W^H is the hidden layer weight matrix and W^X is the weight matrix which is multiplied with the input and is different for each input.

At the end of the network, is an activation function such as a softmax function, which produces the final output, that is a numerical value between 0 and 1 which represents the sentiment of the given sentence.

RNNs, however have an issue; they cannot handle long term dependencies. Just like feedforward neural networks, RNNs update their weight matrices using backward propagation through time(BPTT), that is, for each layer, the error is calculated and sent backwards through the network, and the weight matrices are updated. This can lead to the vanishing gradient problem which results in a point at which the network stops learning.

Hence, we propose to use an LSTM (Long Short-Term Memory) network, which is a more complex and effective RNN which overcomes with the vanishing gradient problem and is hence capable of learning long term dependencies. [6] An LSTM is also composed of a chain of repeating units however unlike in RNNs, each unit consists of four layers. The four components of a unit of an LSTM are; an input gate, a forget gate, an output gate and a new memory container. Figure -- depicts the internal components of an LSTM unit.

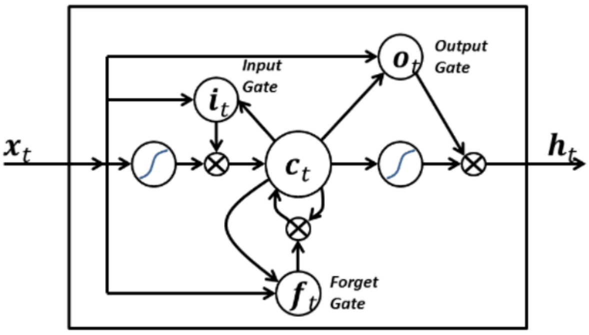

Figure 3. LSTM [6]

The input gate is used to give different amounts of emphasis to different inputs, the forget gate is used to decide which information will continue to persist and which information is unnecessary and can be thrown away, and the output gate is used to determine the final output h_t. With the help of these components, an LSTM can regulate what inputs are important and should impact the final output of a sentence and can eliminate words such as ‘the’ and ‘a’, which have no bearing on the sentiment of a sentence.

V. CONCLUSION

Sentiment analysis is a very useful task for businesses to understand the emotions of their customers regarding their products and services. There are several methods of performing sentiment analysis; deep learning is one such, largely unexplored method. This methodology proposes a deep learning model to classify the Stanford Sentiment Treebank data set and the SAR14 movie review data set. The main steps involved in the proposed methodology are; pre-processing, word embedding and feeding input into a deep neural network architecture. A hybrid architecture of CNNs and RNNs is proposed to achieve optimal accuracy.

There are three types of sentiment analysis; document level, sentence level, and aspect level. Document level sentiment analysis refers to labelling the sentiment of an entire piece of text while sentence level sentiment analysis labels the sentiment of individual sentences. Aspect level sentiment analysis on the other hand identifies the main features in a document and labels the sentiment expressed regarding those features. For example, in a review of a smartphone, an aspect level sentiment analysis model will identify features such as the camera and battery life and assign a label to each feature.

VI. FUTURE SCOPE

The proposed model will perform sentence level sentiment analysis; however, we will modify the model in the future to perform aspect level sentiment analysis. We will also further extend our model and improve its performance by implementing an architecture that uses boosted CNNs. A normal CNN as proposed in the model, takes a fixed length n gram input, however a boosted CNN architecture is composed of a series of CNNs where each CNN has different filters and an Adaboost which will filter the outputs and pick the best result as the final output. This will improve the accuracy of the model.

VII. REFERENCES

[1] Alper Kursat Uysal, Yi Lu Murphey "Sentiment classification: Feature selection based approaches versus deep learning", Computer and Information Technology (CIT), 2017 IEEE International Conference, 21-23 Aug. 2017

[2] Joaqu´ın Ruales, "Recurrent Neural Networks for Sentiment Analysis",

http://joaquin.rual.es/assets/pdfs/rnn_class_project.pdf

[3] Arman S. Zharmagambetov, Alexandr A. Pak, "Sentiment analysis of a document using deep learning approach and decision trees", Electronics Computer and Computation (ICECCO), 2015 Twelve International Conference, 27-30 Sept. 2015

[4] Pallavi Sharma, Nidhi Mishra, "Feature level Sentiment Analysis on Movie Reviews", 2016 2nd International Conference on Next Generation Computing Technologies, 14-16 Oct 2016

[5] Applications of Deep Learning to Sentiment Analysis of Movie Reviews, Report for CS224d: Deep Learning for Natural Language Processing (Stanford), 2015

[6] Abdalraouf Hassan, Ausif Mahmood, "Deep learning for sentence classification", Systems, Applications and Technology Conference (LISAT), 2017 IEEE Long Island

[7] Xi Ouyang, Pan Zhou, Cheng Hua Li, Lijun Liu, "Sentiment Analysis Using Convolutional Neural Network", 2015 IEEE International Conference on Computer and Information Technology, 26-28 Oct. 2015

[8] Abdalraouf Hassan, Ausif Mahmood "Efficient Deep Learning Model for Text Classification Based on Recurrent and Convolutional Layers", 2017 16th IEEE International Conference on Machine Learning and Applications, 18-21 Dec. 2017

[9] Cameron Godbout, "Recurrent Neural Networks for Beginners", https://medium.com/@camrongodbout/recurrent-neural-networks-for-beginners-7aca4e933b82

[10] Christopher Olah, "Understanding LSTM Networks", http://colah.github.io/posts/2015-08-Understanding-LSTMs

[11] Adit Deshpande, "Perform sentiment analysis with LSTMs, using TensorFlow",

https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow