DEEP LEARNING BASED SENTIMENT ANALYSIS FOR RECOMMENDER SYSTEM

Pradeepthi Nimirthi, Venkata Krishna P.

SPM University of Tirupati-India, Department of Computer Science

Corresponding Author: Venkata Krishna P., parimalavk@gmail.com

ABSTRACT: Recommender Systems are used to predict and recommend products based on user preferences. Recommender System is a program that makes use of sentiment analysis technique. Sentiment analysis is one of the popular techniques of the Natural Language Processing (NLP). To perform sentiment analysis task there exists so many approaches, among the approaches we consider the classification task by making use of deep learning models. In this paper we will discuss about the various models that are used for feature generation, which will be provided as input for the sentiment classification. In this paper we also discuss about the sentiment classification task achieved by the linear classifier such as logistic regression and non-linear classifier neural networks such as Convolutional neural network. In this paper we will also discuss about the application of deep learning based sentiment analysis by linear classifier for recommender system on cloud.

KEY WORDS: Recommender system; sentiment analysis; feature; deep learning; linear classifier.

1. INTRODUCTION

Recommender Systems will suggest users to support their decisions in choosing items based on their interests. Now a day's people share their opinions or views on social networking sites such as twitter, face book etc. Building a personalized recommender system on cloud is an interesting task. This paper discuss about building a framework for deep learning based sentiment analysis recommender system (DLSARS) on cloud.

The use of sentiment analysis based recommender system is demonstrated in recent literature. In 2013 P.V. Krishna et al proposed a learning automata based sentiment analysis for recommender system which has been integrated with a cloud platform. This paper proposed a framework called LASA, which is useful in estimating the number of positive or negative responses of a place or item by an individual user [K+13]. In 2017 G. Preethi and P. V. Krishna et.al proposed a recommender system on cloud by applying deep learning concept to sentiment analysis called RDSA framework which was built on 'Recommend Me' App [P+17]. In this paper we try to provide a practical solution to deep learning based sentiment analysis for recommendation on cloud platforms and social networks such as twitter. The sentiment analysis technique used on DLSARS framework on cloud was based on supervised machine learning approach. The DLSARS framework on cloud is used to predict the product reviews based on user preferences such as movies and product reviews.

Generally deep learning models were widely used in the field of pattern recognition and image processing. Now a days deep learning models are widely used to perform Natural Language Processing tasks. Performing sentiment analysis task using deep learning model comes under the branch of machine learning which allows good representation of learning with multiple levels of nonlinear neural networks. Sentiment analysis task was achieved by classifying the task into two levels; first one is providing the input text in the form of features and the later is the classification of the sentiment based on the provided input text features. Generally NLP [PL08, Liu12] techniques use machine learning approaches for classification of linear models such as Support Vector Machines (SVM) or Logistic Regression which takes inputs that are trained over high dimensional sparse feature vectors [Gol16].

Deep learning models use deep neural networks, which are used to achieve the two level tasks. Deep learning models need word embedding as an input feature which is applied for detection and classification task. We can also use neural networks prediction model that is used to learn the word embeddings, which are used for second level tasks like sentiment classification. The input feature that is provided to perform sentiment classification based on neural network based models can be represented as either low dimensional dense vector or high dimensional sparse vector. Deep learning neural network models are classified as non linear classifiers. Representing input feature as dense vector is advantageous over sparse vector because neural networks work well with the dense vector instead of sparse vector. While working with high dimensional sparse vector we can overcome the difficulty with some engineering effort [JZ15a].

2. SYSTEM MODEL

2.1 DLSARS Framework

The main idea behind this model is to collect information from nodes. The collected information from nodes includes latitude and longitude geographic location. We apply deep learning based sentiment analysis on the collected information. We extract the sentiment of the comment either a positive or negative response. This response will be based on the previous history of the user. Based on the previous history we predict the sentiment score of the user. Based on the sentiment score we recommend whether the product review is positive negative or neutral. The cloud platform and social network that is used to collect information from the nodes can be twitter, facebook etc.

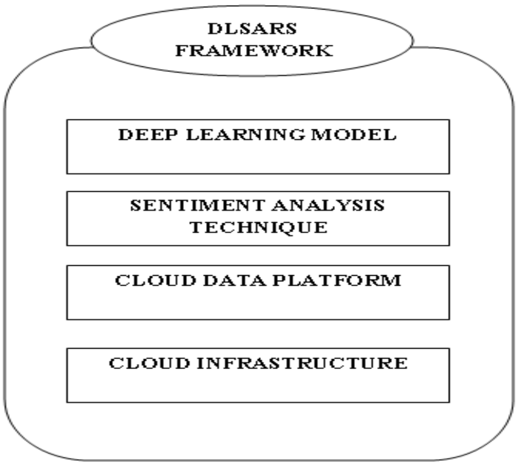

Figure 1. DLSARS Framework

2.2 Deep Learning Model

We propose different deep learning models for feature extraction and classification of learning data and also we will use deep learning model to predict the product response either positive or negative or neutral. Here in this paper we discussed about different models for feature generation and sentiment classification. Application of Deep learning models to the sentiment analysis technique will optimize the performance. Deep learning models include linear and non-linear classifiers with high dimensional sparse vector or low dimensional dense vector.

2.3 Sentiment Analysis Technique

Sentiment analysis technique that applied in this framework includes both feature extraction and classification. This technique suggest different models such as BOW and CBOW models for feature extraction and also suggests different classifiers such as linear and non-linear classifiers. In this paper we get the resultant response as positive or negative with probability factor between 0 and 1. The present technique used for sentiment analysis will also include unigram and bigram approach. The classifiers that take input as the features as high dimensional sparse vector or low dimensional dense vector.

3. FEATURE GENERATION

3.1 Bag of Words Model

For feature generation there exists so many models among them, mainly used is Bag of Words model. This model is used as a tool of feature generation. In this model we represent an order less document representation, where we can count the words that are mattered. Bag of words model is a special case of n-gram model such as Unigram and Bigram. In this model we can characterize or calculate the features of the document from term frequency and normalize the term frequency by the inverse of a document frequency (IDF). This tf-idf will reflect the importance of a word in the document. The main idea to convert the term frequency (TF) and inverse document frequency (IDF) weighting is to transform the values into Vector Space Model (VSM) which increases the performance of the classification systems. We can say that Bag of Words model is a simplified and efficient Vector Space Model. Bag of Words or N-gram model is used to represent the document in a Vector Space Model. Where each vector is represented as a high dimensional sparse vector and each document consists of feature vectors weights are represented as [AJM13].

$$D = \{D_1, D_2, \ldots, D_n\}$$

$$D_i = \{W_1, W_2, \ldots\ldots, W_k\}$$

Where:
D – Set of documents in a Vector Space Model.
n – Total number of Documents.
D_i – i^{th} document that represented as a high dimensional feature vector weight (W) in bag of words model.
The basic n-gram language model is used to assign a probability P(W) to every possible word sequence W i.e. the feature derived from [XR00, PT10] is represented as

$$P(W) = \sum_{i=1}^{v} \left(w_i \mid w_{i-1}\right) \qquad (1)$$

Where: v - Size of the Vocabulary.

The term frequency and inverse document frequency of each high dimensional feature weight vector is represented as follows [M+13].

$$\mathbb{W}_i = TF * IDF \qquad (2)$$
$$TF = tf_i$$

Where:
tf_i– the number of times term i occurs in Document D_i
IDF = $idf_i = \log(n / tdf_i)$
idf_i - Inverse Document Frequency of term i
tdf_i- It is the number of documents that contain term i.

$$BOW(f_1, f_2, \ldots, f_k) = \frac{1}{k}\sum_{i=1}^{k} v(f_i) \qquad (3)$$

Where:
BOW - Bag of Words Model
$f_1, f_2, \ldots, f_k$ - Features of Document D_i
$v(f_i)$ – Sparse vector

$$WBOW(f_1, f_2, \ldots, f_k) = \frac{1}{\sum_i^k \mathbb{W}_i}\sum_i^k \mathbb{W}_i v(fi) \qquad (4)$$

Where: $WBOW$ – Weighted Bag of Words
Here each feature f_i has an associated weight $\mathbb{W}_i$, which indicates the relative importance of feature. Where $\mathbb{W}_i$ could be the TF-IDF score.

3.2 Word Embedding

Word Embedding is a neural network approach, in which we represent the feature vector in a low dimensional space, i.e. we represent the feature vector as a dense vector. The input text feature was represented as distributed representation of text, that can be obtained by various neural network based language models. This approach (Word Embedding) differs due to the removal of non-linear hidden layer to provide additional speed up.
In Unsupervised Word-Embedding algorithm include two new models to represent distributed representation of text such as Continuous Bag of Words model and Continuous Skip-Gram model. These two models are helpful to reduce the computational complexity.

3.2.1 Continuous Bag of Words Model (CBOW)

This model predicts the current word based on the context [M+11]. This model architecture is similar to eed forward Neural Net Language model. This model is similar to traditional Bag of Words model. When Bag of Words and Continuous Bag of Words models were compared, both are used to create input feature vector when the number of features is not known in advance. Both the cases we ignore the order of information. The Bag of Words feature vector is sum or average of all one-hot vectors of the words. Whereas Continuous Bag of Words feature vector is sum or average of all dense vectors instead of sparse vector. Continuous Bag of words model was represented as [Gol16].

$$CBOW(f_1, f_2, \ldots, f_k) = \frac{1}{k}\sum_{i=1}^{k} v(f_i) \qquad (5)$$

Where:
$CBOW$ – Continuous Bag of Words model
$f_1, f_2, \ldots, f_k$ - Features of Document D
$v(f_i)$ – Dense vector

$$WCBOW(f_1, f_2, \ldots, f_k) = \frac{1}{\sum_i^k \mathbb{W}_i}\sum_i^k \mathbb{W}_i v(fi) \qquad (6)$$

Where:
$WCBOW$ – Weighted Continuous Bag of Words.
Here each feature f_i has an associated weight $\mathbb{W}_i$, which indicates the relative importance of feature. Where $\mathbb{W}_i$ could be the TF-IDF score.

3.2.2 Continuous Skip Gram Model

The paper Continuous Skip-Gram model approach is contrast to Continuous Bag of Words Model, where it predicts the surrounding words given the current word.
These two models which are used to represent the distributed representation of text reduce the training complexity because of efficient multithreaded implementation. The process for the input feature vector generation using different models is shown in figure 2.

4. SENTIMENT CLASSIFICATION

The models that we have discussed so far are useful for feature generation which produces high dimensional sparse vector or low dimensional dense vector that is used to refer to the actual input that is fed to the linear classifier or non-linear classifier. In this paper we will discuss about the linear classifier as logistic regression and non-linear classifier as Convolutional neural network for sentiment classification. The process for the sentiment classification using different classifiers is shown in figure 3.

The input feature vector that is fed to the classifier will predict the sentiment. We define the sentiment label of the given text is represented as [M+11]. The sentiment label can be categorical, continuous or multi-dimensional.

$$\hat{S} = f(x) \tag{7}$$

Where:
S– Sentiment polarity of a document $S \in [0, 1]$.

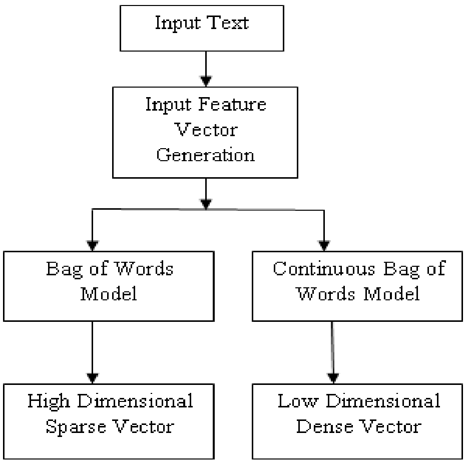

Figure 2: Process for Feature Generation

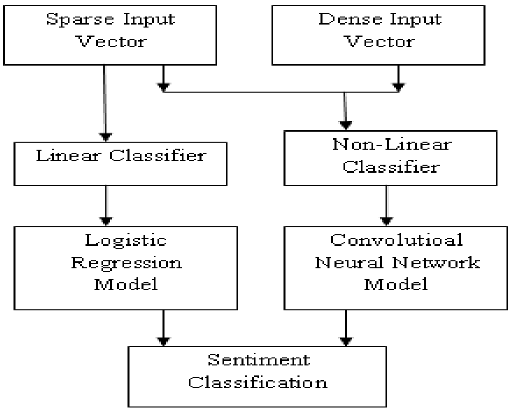

Figure 3: Process showing sentiment classification by linear and non-linear classifiers

4.1 Logistic Regression

The sentiment classification is the process of classifying the polarity of the text either positive or negative. In this model we mainly focus on binary response. When the response i.e. sentiment classification is binary, the linear logistic regression model is often used. Logistic regression is a probability statistical classification type model.
Generally logistic function is defined as from [D+15] will be represented as

$$f(x) = \sigma(t) \tag{8}$$

$$\sigma(t) = \left(\frac{1}{1+e^{-t}} \right)$$

We model the conditional probability using logistic regression as

$$P(S \mid x, \theta) = \sigma\left(\theta^T x\right) = \left(\frac{1}{1+e^{-\theta^T x}} \right) \tag{9}$$

Where x is the feature vector and S is the response and θ are the parameters we wish to learn.

4.2 Convolutional Neural Network

Convolutional neural networks are neural network that are used for processing data that has known grid like topology. To perform Natural Language Processing tasks using Convolutional neural networks, we exploit 1D structure of text data for accurate prediction. A convolutional neural network gives superior classification accuracy because of the non-linearity of the network as well as the ability to easily integrate pre-trained word embeddings. Convolutional neural networks are networks with convolutional and pooling layers, which are useful for classification tasks such as sentiment classification etc. At Convoutional neural networks, Convolutional and pooling architectures are applied to text. Natural Language processing are mainly concerned with one dimension sequence convolutions. Convolution is a specialized kind of linear operation and pooling is an operation in Convolutional neural networks. The Convolution operation is represented as [GBC17] follows.

$$s(t) = (x * w)(t) \tag{10}$$

Where:
∗ - Convolutional operation
s(t) – referred to as feature map
x – referred to as input
w – referred to as the function such as kernel.
When low dimensional dense vector is fed to the Convolutional neural network to predict the sentiment as represented as [Kim14, Shi15].

$$s(t) = c_1 = f(W.x + b)$$

$$\hat{c} = \max\left(c_1, c_2, \ldots\ldots, c_n\right)$$

$$\hat{S} = f(x) = softmax(W\hat{c} + b) \tag{11}$$

Where:

$c_1, c_2, \ldots\ldots c_n$ - Convolution
b - Bias vector
$\hat{c}$ - pooling layer to generate one vector as the output of the filter layer.
When high dimensional sparse vector is fed to the Convolutional neural networks, to predict the sentiment is represented as [JZ15a, JZ15b].

$$\hat{S} = f(x) = \sigma(W.x + b) \qquad (12)$$

Where:
W - Weight matrix
x - Bag of n-gram or bag of word vector
b - Bias vector

5. EXPERIMENTAL RESULTS

The experimental results of our model using linear classifier against other varient models of Convoultional neural networks are shown below [JZ15a, Kim14]. These results show the performance of various classifiers using various techniques by using different feature vectors. All the results based on movie review domain. The below results in table 1 shows the performance of the sentiment analysis technique by using various deep learning models. Below figures 4 and 5 shows the maximum performance of sentiment analysis using linear classifier, Bag of Words (BOW) technique with unigram and bigram approach is 91.27 and 92.09 respectively.

Table 1: Performance of variant models of linear and non-linear classifiers

Technique	Classifier	Feature Vector	Domain	Performance
BOW	Linear Classifier	Sparse vector	Movie reviews	91.27
Bi-gram BOW	Linear Classifier	Sparse vector	Movie reviews	92.09
BOW	CNN	Sparse vector	Movie reviews	86.6
CBOW	CNN-Static	Dense vector	Movie reviews	81.0
CBOW	CNN-random	Dense vector	Movie reviews	81.5

The below Figure 6 shows the visualization of the product iphone8 reviews that are requested from twitter API. The number of tweets that were requested from twitter API was 1000. The visualization shows the probability of positiveness curve with blue line and also it shows positive reviews as above the green dashed line, where as negative reviews as below the red dashed line and neutral reviews between the red and green dashed line.

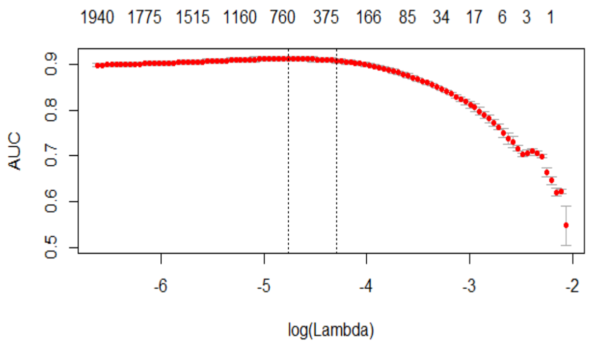

Figure 4. Accuracy of Model Performance using Unigram Approach

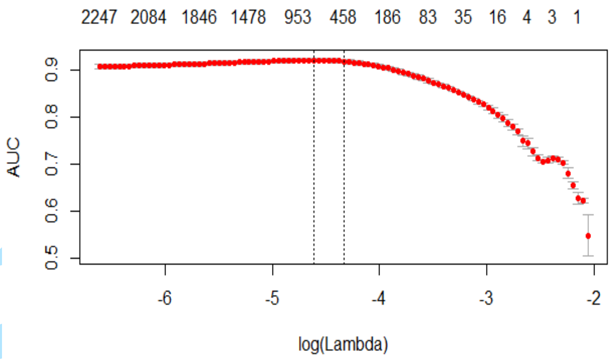

Figure 5. Accuracy of Model Performance using Bigram Approach

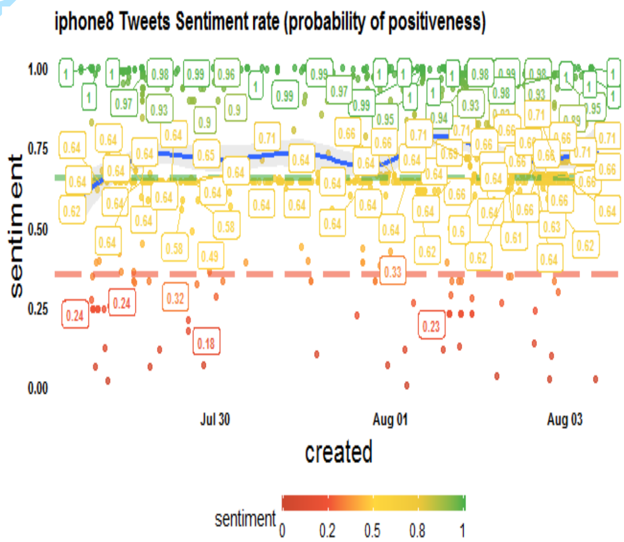

Figure 6. Visualization of Predicted Product Reviews by DLSARS Framework

The below figure 7 shows the predicted accuracy of the product iphone8 reviews that are extracted from twitter API. The vertical dotted line indicates the maximum accuracy that was obtained from the reviews as 89.29.

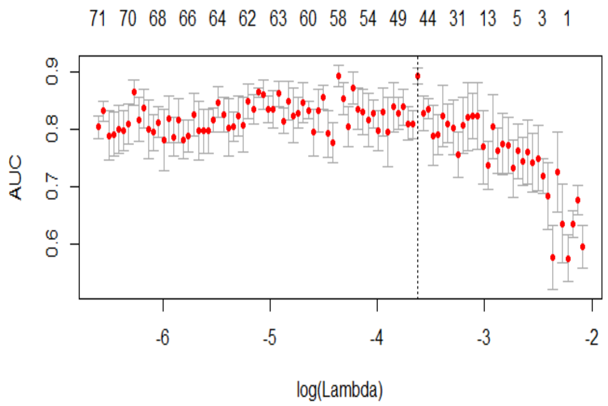

Figure 7. Predicted Accuracy of Product Reviews by DLSARS Framework

6. CONCLUSIONS

In this paper we discussed about the DLSARS framework and application of Deep learning based sentiment analysis for recommender system on cloud. We also discussed about various models such as bag of words or bag of n-gram, Continuous bag of words and skip-gram models to generate or represent features as high dimensional sparse vector or low dimensional dense vector. These feature vectors will be fed as input to the linear or non linear classifiers to classify sentiment polarity. In this paper we also discussed about the linear and non linear classifiers such as logistic regression and Convolutional neural networks respectively for sentiment classification.

REFERENCES

[AJM13] **Alahmadi A., Joorabchi A., Mahdi A. E.** - *A new text representation scheme combining Bag-of-Words and Bag-of-Concepts approaches for automatic text classification*, 7th IEEE GCC Conference and Exhibition *(GCC)*, Doha, pp. 108-113, 2013.

[D+15] **Dawra N., Kovachki N., Lew A., Mills W., Ni X.** - *Sentiment Analysis*, March 16, 2015.

[Gol16] **Goldberg Y.** - *A Primer on Neural Network Models for Natural Language Processing*, J. Artif. Intell. *Res.* 57 (2016): 345-420.

[GBC17] **Goodfellow I., Bengio Y., Courville A.** - *Deep Learning*, Cambridge, MA: MIT Press, 2017.

[JZ15a] **Johnson R., Zhang T.** - *Effective Use of Word Order for Text Categorization with Convolutional Neural Networks*, HLT-NAACL, 2015.

[JZ15b] **Johnson R., Zhang T.** - *Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding*, Advances in neural information processing systems, 2015.

[Kim14] **Kim Y.** - *Convolutional Neural Networks for Sentence Classification*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014.

[K+13] **Krishna P., Misra S., Joshi D., Obaidat M. S.** - *Learning Automata Based Sentiment Analysis for recommender system on cloud*. 2013 International Conference on Computer, Information and Telecommunication Systems, CITS 2013. 1-5.

[Liu12] **Liu B.** - *Sentiment analysis and opinion mining*, Synth. Lect Hum. Lang. Technol. 5 (2012) 1–167.

[M+11] **Maas A., Daly R. E., Pham P. T., Huang D., Ng A. Y., Potts C.** - *Learning Word Vectors for Sentiment Analysis*. ACL, 2011.

[M+13] **Mikolov T., Chen T., Corrado G., Dean J.** - *Efficient Estimation of Word Representations in Vector Space*. CoRR abs/1301.3781, 2013.

[PL08] **Pang L., Lee L.** - *Opinion mining and sentiment analysis*, Foundations and Trends in Information Retrieval, 2 (2008) 1–135.

[PT10] **Paltoglou G., Thelwall M.** - *A Study of Information Retrieval Weighting Schemes for Sentiment Analysis*. ACL, 2010.

[P+17] **Preethi G., P. Venkata Krishna, Obaidat M. S., Saritha V., Yenduri S.** - *Application of Deep Learning to Sentiment Analysis for recommender system on cloud.* CITS2017, 93-97, 2017.

[Shi15] **Shirani-Mehr H.** - *Applications of Deep Learning to Sentiment Analysis of Movie Reviews*, 2015.

[XR00] **Xu W., Rudnicky A.** - *Can artificial neural networks learn language models?*. 202-205, 2000.