Sentiment Analysis: An Empirical Comparative Study of Various Machine Learning Approaches

Swapnil Jain, Shrikant Malviya, Rohit Mishra, Uma Shanker Tiwary
Department of Information Technology
Indian Institute of Information Technology Allahabad
Allahabad-211012 (Uttar Pradesh)
{j.swapnil2050,shrikant.iet6153,rohit129iiita,ustiwary}@gmail.com

Abstract

The aim of this paper is to experiment with different machine learning approaches to predict/classify the sentiment on various available sentiment corpuses named as Subjectivity v1.0 corpus, IMDB movie review corpus, Rotten Tomatoes (RT) Movie Reviews corpus, Twitter sentiment dataset. Variants of Naive Bayes (NB) and Support Vector Machines (SVM) have been often used for text categorization as baseline. In this paper, we have tried to show that how embodying bigram and trigram features with Logistic Regression (LR), Mutinomial Naive Bayes (MNB) and Support Vector Machine (SVM) show significant improvent in the sentiment anlaysis. Another observation we obtained is that LR outperforms the MNB and SVM in both large as well as short (snippets) sentiment text when sentiment classes are limited to two/three. Furthermore, when the sentiment analysis task turn into a kind of multi-class classifiction instead of binary on large corpora, deep learning become dominant. We obtained testing accuracy of 96.6% and training accuracy of 98.8% on IMDB corpus by LR with unigram+bigram+trigram feature variant. Similarly, for Subjectivity v1.0 and twitter corpus, the same model returns better accuracy. But on the multi-class RT movie reviews corpus, Deep learning based proposed architecture-3 of type Extended-Convolution Neural Network (E-CNN) outperforms others.

1 Introduction

Recently, the field of Opinion Mining and Sentiment Analysis has enticed many researchers around the globe due to its capability of delivering valuable informative applications. People's opinion and reviews can play a crucial role in making decision's and choosing among multiple options when those choices are related on valuable resources for example expenditure of time and money to buy products as well as services. These information mostly sourced from *social web* through several forums, blogs and social networking websites. However, Due to its heterogeneous and unstructured nature, this information is not directly *machine processable*. Thus, it set the reason for the emergence of *Opinion Mining* (OM) and *Sentiment Analysis* (SA) as a prominent area of research. Both the keywords are commonly used interchangeably to denote the same meaning. However, some researchers believe, both aim to solve two slightly different problems. According to (Tsytsarau and Palpanas, 2012), OM determine whether a piece of text contains opinion or not, a problem that considered as *subjectivity analysis*. On the other hand, SA's task is to measure the polarity of text i.e. positive or negative.

Polarity classification is known to be very basic task of OM and SA. Polarity classification as the name signifies, classify a piece of text related to opinion on a particular issue into two sentimental opposite class. Moreover, it also helps in identifying pros and cons expressions of customer reviews which make the product evaluation and customer interest assessment more credible.

In the present scenario, sentiment analysis and opinion mining depends on the vector extraction of a piece of text in order to represent it's most salient and important features. These features representing a specific patterns-set help in determining the proper sentiment/opinion class. Term frequency, presence and tf-idf[1] are commonly used features.

[1]tf-idf, short for term frequency-inverse document frequency

S Bandyopadhyay, D S Sharma and R Sangal. Proc. of the 14th Intl. Conference on Natural Language Processing, pages 112–121,
Kolkata, India. December 2017.

In this research, we study the empirical effects related to several variants of LR, MNB, SVM on various available sentiment datasets. However, these approaches are already used enormously in text categorization, their performance varies due to inherent variability in features, datasets and model used. Through a set of experiments done on many datasets, we tried to show that the better selection of variants in many cases outperform the recent published state-of-the-art.

2 Related Work

Sentiment analysis field of research has been studied and employed widely since last two decades. SA systems have been implemented through different levels of analysis, such as word level e.g., (Qiu et al., 2009), the attribute level e.g., (Mei et al., 2007), the concept level e.g., (Cambria and Hussain, 2012), the sentence or clause level e.g., (Wilson et al., 2004) and finally the document level e.g. (Pang et al., 2002).

The Sentiment analysis is also understood as a task of determining the sentiment orientation of a given textual unit distinguished into two or more classes. Hence, the task of sentiment classification has also been implemented for different number of classes such as binary (e.g. positive/negative classification), ternary (e.g. positive/negative/neutral), n-ary (e.g. 1-5 star labelling) (Rui et al., 2013).

In general, the SA approaches can be classified into two main categories, the dictionary based approaches and other one is machine learning based approaches (Saad, 2014). Dictionary-based approaches are also known as lexical-based approaches that utilize a set of predefined set of sentiment dictionaries to identify the sentiments in a given text. At the starting, most of the work in the field of sentiment analysis was focused only on the dictionary-based approaches. On the other hand, machine learning approaches are become popular in recent years which work through constructing a classifier trained on manually annotated corpus to discriminate the sentiments of a given text.

Likewise, Decision Trees (DT), Naive Bayes (NB), Support Vector Machine (SVM), Neural Network (NN) and Maximum Entropy (ME) are the common set of supervised learning approaches, applied in sentiment classification (Medhat et al., 2014). Each type of approaches have its own pros and cons. For example, the dictionary-based approaches suffers from the limitation of highly domain-orientedness. Likewise, the machine learning approaches also require a significant human effort in order to annotate a substantial number of examples for training a classification model first.

OM and SA are in real, non-trivial and challenging problem, spanned over many areas and applications. However, a significant number of studies have been done in this field since past one decade, still much remained to be explored in order to build robust real-life applications. It has been observed that the problem of differentiating subjective with objective instances of sentiment is more difficult than the later polarity classification (Molina-González et al., 2013). Therefore, any improvement made on the field of subjective classification will put positive impact on sentiment classification. In the past, it has been done not only using machine learning (Wang et al., 2011; Pang and Lee, 2004) but lexicon-based approaches are also been adapted (Banea et al., 2014; Xuan et al., 2012). A glimpse of some subjective classification results obtained by the researchers in the past on Pang and Lee (Pang and Lee, 2004) corpus are shown in Table 1.

Sometimes, sentiment classification is related to identification of polarity of a piece of text whether it is showing positive, negative or neutral sentiment (Wilson et al., 2005; Turney, 2002). Therefore, sometimes sentiment classification is also called as polarity determination. Polarity determination has been tried on product reviews, blogs, micro blogs, news articles and forums. It's been observed that such texts are full of non-linguistic content e.g. abbreviations, noisy texts. Hence, it is required to use high level of preprocessing and more intelligent analytical techniques in order to extract most important discriminating patterns. These micro-blogs are proved to be more prominent and useful objects for many applications such as inferring opinion in social networks, twitter mood prediction, social advertising over micro-blogs and user-interest prediction in micro-blogging etc. (Maks and Vossen, 2012; Bollen et al., 2011; Bao et al., 2013; Li and Shiu, 2012).

3 Corpora Description

The goal of this paper is to deliver a comparative study of various machine learning approaches on different datasets. A number of relevant benchmark datasets are used and analysed with sev-

Authors	Data Split	Classifier Models	Cross Validation	Feature Selection	Baseline Accuracy (%)	Best Accuracy (%)
(Pang et al., 2002)	700 Positive 700 Negative	NB, ME, SVM	3-fold	unigrams presence	-	82.90
(Pang and Lee, 2004)	1000 Positive 1000 Negative	NB, SVM	10-fold	unigrams presence	87.15	87.20
(Mullen and Collier, 2004)	700 Positive 700 Negative	Hybrid SVM	10-fold	PMI,Turney, Osgood,Lemmas	83.50	87.00
(König and Brill, 2006)	1000 Positive 1000 Negative	Text Pattern + SVM, SVM	5-fold	unigrams bigrams	87.50	91.00
(Abbasi et al., 2008)	1000 Positive 1000 Negative	Genetic Algorithm Genetic Algorithm with SVM	10-fold	POS/Words n-grams Punctuation	87.95	91.70
(Prabowo and Thelwall, 2009)	1000 Positive 1000 Negative	Hybrid (Rule + Statistical and SVM)	5-fold	term frequency term presence	87.30	87.30

Table 1: Recently published Results in the literature on various versions of (Pang et al., 2002) movie review dataset.

Dataset	Type	No. of Textual Units	Positive	Somewhat Positive	Negative	Somewhat Negative	Neutral
Subjectivity v1.0 Corpus	Snippets of Movie Reviews	10662	5331	-	5331	-	-
IMDB Dataset	Movie Reviews	50k	25K	-	25K	-	-
Twitter Sentiment Dataset	Tweets on Flight Service	14640	2363	-	9178	-	3099
Rotten Tomatoes Dataset	Movie Reviews	156060	9291	32681	7565	27325	79198

Table 2: Statistics of the datasets used in this paper.

eral methods in order to find their individual characersitics towards the various approaches. We have considered four different corpora in order to perform the experiments: (1) Rotten Tomatoes Dataset (Kaggle-Competitions, 2017), (2) Subjectivity v1.0 Corpus (Pang and Lee, 2005), (3) IMDB Movie Review Dataset (Maas et al., 2011) and (4) Twitter Sentiment Dataset (Twitter-Crowdflower, 2017).

3.1 Rotten Tomatoes Dataset

This is one of the renowned corpus for statistical sentiment analysis on the collection of movie reviews prepared by Pang and Lee (Pang and Lee, 2004). The corpus[2] was prepared in order to classify movie reviews as positive or negative that are collected from the IMDB.com (Internet Movie DataBase). Initially, the corpus was consisted of 2000 full length reviews, 1000 each of positive as well as negative. Later, the dataset transformed to carry reviews of sentiments scaled in range [1-5]. Recently, a contest was hosted on (Kaggle-Competitions, 2017) with a huge corpus of movie reviews taken from rotten tomattoes on 5-star rating scale. We used this updated large corpora in this paper to see the the difference in results of various approaches. As the collected reviews are classified according to the rating system in terms of 5-star, mulit-class machine classification approaches are applied to develop a robust sentiment classification model.

3.2 Subjectivity v1.0 Corpus

A *sentence polarity dataset*[3] has been created by Pang & Lee, consists of 5331 of each positive as well as negative short movie reviews "snippets" (compulsorily one single long sentence) extracted from `www.rottentomatoes.com` (RT-s) (Pang and Lee, 2005). The aim of collecting this dataset is to understand the sentiment analysis paradigm on short subjective reviews and objective plot summaries instead of considereing the complete large reviews. Each snippet in the corpora is marked as "positive" if it is labelled "fresh" in `www.rottentomatoes.com` and the other snippets which are marked with "rotten" are considered to be negative reviews.

3.3 IMDB Review Dataset

Another movie review dataset has been collected by Andrew Maas at Stanford, sourced from IMDB (Maas et al., 2011). The dataset consists of 50,000 reviews in total, 25,000 of each positive as well as negative sentiments, conditioned on no more than 30 reviews from one movie. The reviews are distributed evenly into positive and negative classes so that the random selection will result in 50% accuracy. As movie reviews in IMDB are scored from 1 to 10 scale, the selected negative reviews are considered if its score is ≤ 4 out of 10 and for the positive reviews the threshold is set to ≥ 7

[2]The dataset is freely available at `www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz`

[3]The dataset is freely available at `www.cs.cornell.edu/people/pabo/movie-review-data/rt-polaritydata.tar.gz`

out of 10. Other reviews (neutral reviews) are not considered in this dataset.

3.4 Twitter Sentiment Dataset

This dataset originally came from crowd flowers library [4] (Twitter-Crowdflower, 2017). The dataset was generated through undertaking the sort of complaints received by each airline entirely by major U.S. air carrier customer service. The dataset includes tens of thousands of tweets as mentioned in the table 2, their respective carriers, the positive, negative, and neutral sentiment. This is a manually labelled corpus. In the process of corpus generation, users were asked to manually label the tweets as positive, negative or neutral with reasons of late flight, fast service etc.

4 Classification Models for Sentiment Analysis

4.1 Dataset Pre-processing & Feature Extraction

Data pre-processing is necessary task for sentiment analysis as it performs the process of cleaning and preparing text to be suitable as input to classification models (Haddi et al., 2013). Most of the sentiment dataset are made of the content extracted from websites e.g. Movie Reviews websites, product opinion websites, tweets from twitter etc. They all contain usually lots of noise and uninformative parts such as HTML tags, advertisements and scripts which needed to be removed before sending them for the classification. In order to prepare datasets for applying various machine learning approaches, we have designed a set rules for removal of the noise and uninformative parts i.e. HTML tags, rating indicators etc.

For all datasets, similar steps of pre-processing methods are undertaken. Following steps are followed for the same:

- Removing URL and getting data inside HTML Tag.
- Removing Repeating Characters, i.e. looove = love.
- Replacing emoticons with word happy and sad
  ':D' ':)' ':P' ';)' → happy
  ':(' ' ;(' ':—' → sad
- Replacing marks, ? → qmark , ! → exmark
- Removing stop words and replacing words like (don't → do not) or (thx/thnx → thanks) etc.

Feature Engineering is an important part of text analytics where features are extracted from text. First comes bag of words, a model where words are stored like the elements of a set with no word order or specific grammar known. Second is about use of different encodings. It states how the text could be represented in the form of vectors where the length of the vector is generally considered as the length of vocabulary i.e. the number of distinct words. First comes the very basic Count Encoding which is drawn from the frequency of a word, kept in the vector form. Similarly, the tf-idf encoding deals with constructing vectors of tf-idf weight of the words. Likewise, vector generation can also use ngrams and word-embeddings as features. Under ngram feature space, a single word is known as unigram, a sequence of two and three words are called bigrams and trigrams correspondingly.

Recently, word embeddings become top-notch in order to avail the use of dense or continuous vectors. Its main benefit arguably is that it does not require expensive annotation, instead it can be derived from large unannotated corpora that are readily available. Pre-trained embeddings can then be used in downstream tasks that use small amounts of labelled data. Various Transformations are there for use of word embeddings in a sentence i.e. mean transformation, image transformation. If each word in a sentence will have n embeddings, its mean transformation would be the mean of all the n embeddings. Thus, this will give rise to the feature vector of same length as the length of sentence. On the other hand, if we consider the length of embeddings n and feature vector length m, $[n * m]$ order can be considered as a gray scale image where every element represents pixel intensity and thus it can be feed into a convolution neural network or any other machine learning model as an image.

4.2 Support Vector Machines (SVM)

Support vector machines (SVM) has been applied in this work in order to classify the text units in a set of pre-defined sentiment classes. The algorithm got its name from the fact that it used to find those samples (support vectors) which find the widest frontier between the positive and negative samples in the feature space through demarcating

[4] The dataset is freely available at www.crowdflower.com/data-for-everyone/

those samples (support vectors). Due to its several advantages such as robustness in high dimensional space, versatility to any type of features, highly suitable for linear separable data and robust even when the data is sparsely distributed in the feature space, SVM become suitable to be applicable in many text categorization problems, motivated to be used in the SA. This has been proved by achieving good results on application of SVM in opinion mining and shown that it has overcome other machine learning techniques (OKeefe and Koprinska, 2009). A comparative study of several variants of SVM with other approaches are discussed briefly in the next section Experiments & Results.

4.3 Multinomial Naive Bayes (MNB)

Bayes Theorem based techniques that assumes independence among events/predictors are considered to be Naive Bayes approaches. In simple terms, one feature is not related to any other features, this is the general idea behind the working nature of Naive Bayes. Because of its less time complexity, this model is faster and can be easily used for large datasets. With the power of simplicity in hand, it is also known to outperform even highly complex classification models in many cases (Saad, 2014).

In the Multinomial variation of Naive Bayes, each textual data d is considered as a bag of tokens with each entry in it t_i representing the occurrence of a token or its tf-idf value or any other weight score (Wang and Manning, 2012). Therefore, d can be shown as a vector $\vec{x} =< x_1, x_2, ..., x_n >$, in which each x_i is bound to show the weight of t_i occured in d. Furthermore, each text unit d of a particular class c is considered to be the outcome of selecting individually $|d|$ tokens from F with replacement where each t_i has probability $p(t_i|c)$. Hence, $p(\vec{x}|c)$ is represented by following multinomial distribution:

$$p(\vec{x}|c) = p(|d|) \cdot |d|! \cdot \prod_{i=1}^{m} \frac{p(t_i|c)^{x_i}}{x_i!} \quad (1)$$

here, a common assumption is followed that $|d|$ does not depend on the class c. This is a method that has shown a significant improvement when combined with a combination of unigram, bigram and trigram.

4.4 Logistic Regression (LR)

We now look at the application of another algorithm for sentiment analysis named logistic regression (Wang and Manning, 2012). In terms of classifiers, logistic regression belongs to the exponential or log-linear classifiers family. Like other linear classifiers such as Naive Bayes, it also extracts a set of weighted features from the input, combining them linearly preceded by taking logs. In a more general way, logistic regression is represented by a classifier that classifies a data in two classes.

The most fundamental difference between Naive Bayes and Logistic Regression is that the Naive Bays is a generative classifier while the Logistic Regression is a discriminative classifier (Jurafsky and Martin, 2014). Naive Bays classifier is based on the concept that it probabilistically chooses which output label c is to be assigned to an input x through maximizing $p(c|x)$. It is perceived directly, Naive Bayes classifier used to estimate the best c indirectly on the basis provided likelihood $p(x|c)$ and prior class probability $p(c)$:

$$\hat{c} = \underset{c}{\operatorname{argmax}}\, p(c|x) = \underset{c}{\operatorname{argmax}}\, p(x|c)p(c) \quad (2)$$

Although LR differs in terms of estimating the probabilities, it is still similar to NB as being a linear classifier. LR estimates the term $p(c|x)$ through extracting a set of features from the provided input followed by fusing them linearly with weight vector (dot product) and then putting this combined value to a function. The beauty of exponential function for generating positive outcome, is used as being an applied function here. In general, the basic Logistic Regression formula for estimating the $p(c|x)$ is:

$$p(c|x) = \frac{1}{Z} exp\left(\sum_{i}^{N} w_i f_i(c, x)\right) \quad (3)$$

The denominator in the above equation Z is normalization factor which converts a exponent value to its probability. If vectors are represented by N values, the final equation of calculating the probability of x being of class c through LR:

$$p(c|x) = \frac{exp\left(\sum_{i=1}^{N} w_i f_i(c, x)\right)}{\sum_{c \in C} exp\left(w_i f_i(c', x)\right)} \quad (4)$$

A form of linear regression where the value which we want to predict i.e. c takes the discrete amount which further can be used as the label for a class. The cost function to estimate parameters for logistic regression for binary classification which we intend to minimize is given as follows:

$$J(f) = -\frac{1}{m}\Bigg[\sum_{i=1}^{m} c^{(i)} \log p(c^{(i)}|x^{(i)}) + (1-c^{(i)}) \log(1-p(c^{(i)}|x^{(i)}))\Bigg] \tag{5}$$

Where, m is the number of sample, x is the predictor. Since c here always belongs to either 0 or 1. The strategy used for multiclass classification we used is one versus all where only one class is considered while classification of the rest considered to be zero. Its been shown in the table later under Experiments & Results section that LR proved to be far better than MNB and SVM when it includes bigram and tigram based features.

4.5 Extended Convolution Neural Network (E-CNN)

In this section, we discuss a extended version of convolution neural network (E-CNN), a variant of the CNN architecture used by (Lan et al., 2016). We have deployed this multi-channel variant of CNN, E-CNN (Extended-CNN) in order to capture both semantic as well as sentiment information. In the E-CNN architecture, a sentence of length n with each contained word w_i represented by corresponding k-dimensional vector $w_i \in \mathbb{R}^k$ to the i-th word. Hence, a sentence of length n with added necessary padding if needed is supposed to be represented as

$$w_{1:n} = w_1 \oplus w_2 \oplus ... \oplus w_n, \tag{6}$$

here, $\oplus$ represents a binary operator of concatenating its two operands. Hence, the symbol $w_{1:n}$ refers to concatenated string of n vectors $w_1, w_2, ..., w_n$. Further, the convolution, a dot product operation, filters out a set of features and properties from the input through applying a $filter$ m $\in \mathbb{R}^{hk}$ window of size, say h words, where k is the dimension size of the word vector. In other words, the goal of convolution layer is to generate a feature map c ($c \in \mathbb{R}^{n-h+1}$) like $[c_1, c_2, ..., c_{n-h+1}]$ for input sentence s, where each term c_j is estimated through dot product of convolution filter m with h word vectors ending at word w_j (i.e., $w_{j-h+1:j}$):

$$c_j = f(m^T w_{j-h+1:j} + b) \tag{7}$$

where f is a non-linear activation function such as hyperbolic tangent (Tanh), rectified linear unit (ReLU) and $b \in \mathbb{R}$ is a biased term which allows the activation function to be shifted to left or right for successful learning. Likewise, all the filters convolutes individually to each possible window of the words in the sentence $w_{1:h}, w_{2:h+1}, ..., w_{n-h+1:n}$ in order to generate a $featuremap$

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

Each filter tries to identify only one type of feature. Hence, in order to capture multiple features, CNN models generally employ multiple filters by varying the windows sizes or using the same filter with random initialization each time.

Further, in order to capture the necessary information from each feature map c, several $pooling$ methods have been presented such as averaged pooling (i.e., $\hat{c} = \frac{1}{h}\sum_{i=1}^{h} c_i$) or max-over-time pooling (i.e., $\hat{c} = max(c_i)$). We have used max-over-time pooling operation on the feature map in order to take the maximum value $\hat{c} = max(\mathbf{c})$ as a feature for the corresponding filter. Natural idea to use this pooling operation is to capture the most important feature, nothing but the highest value, for each feature map. The value obtained as features ($z = [\hat{c_1}, \hat{c_2}, ..., \hat{c_k}]$) after pooling are forwarded into a $softmax$ layer:

$$p(y = l|z; \theta) = \frac{e^{z^T.\theta_l}}{\sum_{k=1}^{K} e^{z^T.\theta_k}} \tag{9}$$

which estimates the probability distribution over predefined labels l. Nevertheless, in order to adjust the weights of layers, the parameters in CNN model (i.e. m, f, b, θ) are fine-tuned via back-propagation method.

An experiment is done with two *channels word vectors*, one's job is to capture unsupervised semantic information and another's task is to extract sentiment details from the input. The first, semantic channel is kept static throughout the training and second, sentiment channel is fine-tuned via back-propagation (Kim, 2014).

Model	Training Accuracy	Testing Accuracy
LR Unigram	92.5	91.4
LR Bigram	88.9	87.5
LR Unigram + Bigram	93.3	92.6
LR Unigram + Bigram + Trigram	**98.8**	**96.6**
NB Unigram	91.3	90.2
NB Bigram	92.3	90.3
NB Unigram + Bigram	93.2	90.2
NB Unigram + Bigram + Trigram	**94.4**	**93.6**
SVM + Unigram	85.3	83.4
SVM + Bigram	79.0	77.5
Mean Embeddings + SVM	85.2	84.5
Mean Embeddings + LR	84.2	83.0

Table 3: Accuracy chart of various approaches on IMDB corpus.

5 Experiments & Results

Support Vector Machines are used with different kernels for classification and also in Logistic Regression; we use regularization to penalize the weights to prevent over-fitting. For E-CNN different approaches are taken like changing the filter sizes, pooling layers are also used, changing the number of hidden layers etc. Among various architectures of convolution networks major ones which give promising results, are listed as follows:

- **Architecture-1**
  - Convolution Layer 1D Receptive Field 3x1, Feature Maps 100, Activation relu
  - Convolution Layer 1D Receptive Field 4x1, Feature Maps 100, Activation relu
  - Max Pooling Layer 1D Receptive Field 3x1, Activation relu
  - Fully Connected Layer Neurons 100, Activation relu
  - Fully Connected Layer Neurons 50, Activation sigmoid
  - Output Layer
- **Architecture-2**
  - Convolution Layer 1D Receptive Field 2x1, Feature Maps 150, Activation relu
  - Max Pooling Layer 1D Receptive Field 3x1, Activation relu
  - Convolution Layer 1D Receptive Field 3x1, Feature Maps 150, Activation relu
  - Max Pooling Layer 1D Receptive Field 3x1, Activation relu
  - Fully Connected Layer Neurons 200, Activation relu
  - Fully Connected Layer Neurons 100, Activation sigmoid
  - Fully Connected Layer Neurons 50, Activation sigmoid
  - Output Layer
- **Architecture-3**
  - Convolution Layer 2D Receptive Field 3x3, Feature Maps 100, Activation relu
  - Convolution Layer 2D Receptive Field 3x3, Feature Maps 150, Activation relu
  - Flatten Layer
  - Fully Connected Layer Neurons 100, Activation relu
  - Fully Connected Layer Neurons 100, Activation relu
  - Fully Connected Layer Neurons 64, Activation relu
  - Fully Connected Layer Neurons 10, Activation sigmoid
  - Output Layer
- **Architecture-4**
  - Convolution Layer 2D Receptive Field 5x5, Feature Maps 100, Activation relu
  - Max Pooling Layer 2D Filter Shape 2x2
  - Convolution Layer 2D Receptive Field 4x4, Feature Maps 150, Activation relu
  - Max Pooling 2D Filter Shape 2x2
  - Flatten Layer
  - Fully Connected Layer Neurons 100, Activation relu
  - Fully Connected Layer Neurons 64, Activation sigmoid
  - Output Layer

On the basis of various feature combinations, many possible variants of SVM, MNB and LR such as have been investigated, but only those are mentioned in the tables 3, 4, 5 and 6 which deliver good results. As per the experiments done on the IMDB movie review corpus, It is observed that combination of Unigram, Bigram and Trigram features provide more accurate classification results. Table 3 supports the observation. Both Classes positive/negative are well-classified, but the sentences which were misclassified are mostly related to sarcasm or confusing for human perception. For example -

Predicted Negative but Marked Positive → "You are a total idiot if u dont watch this movie. You are wasting your time on this planet." (Sarcasm)

Model	Training Accuracy	Testing Accuracy
Logistic Regression Unigram	91.9	90.7
Logistic Regression Bigram	82.6	80.9
Logistic Regression Unigram + Bigram + Trigram	**99.7**	**96.4**
Naive Bayes Unigram	94.2	92.9
Naive Bayes Bigram	86.7	84.9
Naive Bayes Unigram + Bigram + Trigram	**97.1**	**95.2**
SVM + Unigram	82.4	82.1
SVM + Bigram	56.2	55.8

Table 4: Accuracy chart of various approaches on Subjectivity v1.0 corpus.

Model	Training Accuracy	Testing Accuracy
Naive Bayes + Count Encoding + unigrams + bigrams	80.3	5-fold
Naive Bayes + Tf-Idf Encoding + unigrams + bigrams	75.3	7-fold
Logistic Regression + Count Encoding + unigrams + bigrams	**82.1**	**5-fold**
Logistic Regression + Tf-Idf Encoding + unigrams + bigrams	81.5	7-fold

Table 5: Avg. Cross Validation Accuracy chart of various approaches on Twitter dataset.

Positive but Marked Negative → "This movie makes me wonder what I am doing on earth wasting time, doing nothing, Ohh Man, What the hell." (Confusing even for human perception)

With reference to published results on subjective v1.0, a sentiment corpus consists of snippets, short reviews, the results presented in this paper is more accurate. Moreover, it also shows, the capability of SVM is better in classification of long reviews. But for the short reviews or snippets as subjective corpus, Logistic regression and Naive Bayes are more accurate and robust. Addition of bi-grams improves the performance significantly as shown in Table 4. After the inclusion of trigram again improve the performance a bit more. Both LR and MNB with unigram, bigram and trigram features provides 96.4% and 95.2% accuracy respectively as shown in Table 4.

For the sentiment analysis experiment on twitter corpus, a number of encoding considered to draw feature set in order to apply some supervised learning methods. Here also, feature vectors are constructed out of various possible combination of unigrams, bigrams with individual count encoding and tf-idf encoding. Out of all combination, only those are shown here which draw significantly better result. The accuracy measurement is done on the set environment of 5-fold and 7-fold avg. cross-validation. The overall average accuracy is obtained 82.1% as shown in the Table 5 through logistic regression in combination with count encoding and unigram+bigram.

The Rotten Tomato Dataset is a very large movie review corpus composed of 156,060 sentences rated under 5-star rating scheme in Negative, Somewhat Negative, Neutral, Somewhat Positive, Positive categories. We divided the overall corpus into a ratio of 7:2:1 for training, test and cross-validation set. The problem of sentiment analysis now turned from binary classification to multi-category classification which make it difficult for the above implemented models to be incorporated here. This is the reason, the accuracy of some linear approaches such as SVM and LR start declining. Therefore, deep learning is undertaken to see the difference. Four different architectures are devised empirically which show better accuracy compare to SVM and LR as shown in Table 6.

It is clear from the above discussion that logistic regression works better on the datasets like imdb, subjectivity v1.0 and twitter where the sentiment classes are limited to two/three and the corpus is build of short statements/reviews. But for the large datasets like Rotten Tomato corpus which consists of millions of texts divided into many sentiment classes, a better model is required robust enough to capture and support entire feature set necessary

Model	Training Accuracy	Testing Accuracy
Mean Embeddings + SVM	52.4	52.0
Count Encoding + LR	61.4	60.5
TfIdf Encoding + LR	63.3	62.5
Architecture-1	58.3	56.2
Architecture-2	59.1	57.5
Architecture-3	**68.4**	**66.7**
Architecture-4	65.1	63.4

Table 6: Accuracy chart of various approaces on Rotten Tomatoes Dataset.

for the classification. For the twitter dataset, only unigrams and bigrams with count encoding give the better results.

As it can be seen, mean transformation of embeddings does not play major role in sentiment analysis whereas image transformation of embeddings achieve the best result among all other classifiers. It is not worth denying that mean transformation is not that good for representation of embeddings as feature vector. Many other transformations for embeddings are there like median, mode, tf-idf but still the combination of convolution neural network with image transformation of embeddings beats them all. So embeddings are quite useful if used wisely.

6 Conclusion

In this paper, we performed a set of experiments to capture the residing variation in various sentiment datasets such as short or long texts and binary vs multi-class classification variations. For this, we analyzed various renowned models for classification and also various architectures for Convolutional Neural Network on all possible datasets ranging from short reviews/snippets to long documents. For each type, a list of best performing models are shown. We observe that for short texts and/or binary classification LR models beat all other models with certain features. In contrast, for long texts like the rotten tomatoes datasets, logistic regression is shown to give accuracy of 62.47% but in order to achieve better accuracy we included the use of word embeddings as an image and feed it into the convolution neural network (proposed architecture-3) where we achieve greater accuracy of 66.70%. Furthermore, for multi-class dataset like rotten tomatoes dataset, based on the analysis of confusion matrix, a better feature set selection and corresponding model enhancement related problems can be considered for future work.

References

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)* 26(3):12.

Carmen Banea, Rada Mihalcea, and Janyce Wiebe. 2014. Sense-level subjectivity in a multilingual setting. *Computer Speech & Language* 28(1):7–19.

Hongyun Bao, Qiudan Li, Stephen Shaoyi Liao, Shuangyong Song, and Heng Gao. 2013. A new temporal and social pmf-based method to predict users' interests in micro-blogging. *Decision Support Systems* 55(3):698–709.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science* 2(1):1–8.

Erik Cambria and Amir Hussain. 2012. *Sentic computing: Techniques, tools, and applications*, volume 2. Springer Science & Business Media.

Emma Haddi, Xiaohui Liu, and Yong Shi. 2013. The role of text pre-processing in sentiment analysis. *Procedia Computer Science* 17:26–32.

Dan Jurafsky and James H Martin. 2014. *Speech and language processing*, volume 3. Pearson London.

Kaggle-Competitions. 2017. Kaggle: Sentiment analysis on movie reviews. https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews/data/. [Online; accessed 22-March-2017].

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Arnd Christian König and Eric Brill. 2006. Reducing the human overhead in text categorization. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 598–603.

Man Lan, Zhihua Zhang, Yue Lu, and Ju Wu. 2016. Three convolutional neural network-based models for learning sentiment word vectors towards sentiment analysis. In *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, pages 3172–3179.

Yung-Ming Li and Ya-Lin Shiu. 2012. A diffusion mechanism for social advertising over microblogs. *Decision Support Systems* 54(1):9–22.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages

142–150. http://www.aclweb.org/anthology/P11-1015.

Isa Maks and Piek Vossen. 2012. A lexicon model for deep sentiment analysis and opinion mining applications. *Decision Support Systems* 53(4):680–688.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* 5(4):1093–1113.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*. ACM, pages 171–180.

M Dolores Molina-González, Eugenio Martínez-Cámara, María-Teresa Martín-Valdivia, and José M Perea-Ortega. 2013. Semantic orientation for polarity classification in spanish reviews. *Expert Systems with Applications* 40(18):7250–7257.

Tony Mullen and Nigel Collier. 2004. Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*. volume 4, pages 412–418.

Tim OKeefe and Irena Koprinska. 2009. Feature selection and weighting methods in sentiment analysis. In *Proceedings of the 14th Australasian document computing symposium, Sydney*. Citeseer, pages 67–74.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, page 271.

Bo Pang and Lillian Lee. 2005. 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*. Association for Computational Linguistics, pages 115–124.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pages 79–86.

Rudy Prabowo and Mike Thelwall. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics* 3(2):143–157.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *IJCAI*. volume 9, pages 1199–1204.

Huaxia Rui, Yizao Liu, and Andrew Whinston. 2013. Whose and what chatter matters? the effect of tweets on movie sales. *Decision Support Systems* 55(4):863–870.

Farag Saad. 2014. Baseline evaluation: an empirical study of the performance of machine learning algorithms in short snippet sentiment analysis. In *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*. ACM, page 6.

Mikalai Tsytsarau and Themis Palpanas. 2012. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery* 24(3):478–514.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 417–424.

Twitter-Crowdflower. 2017. Crowdflower: Sentiment analysis on twitter dataset. `https://www.crowdflower.com/data-for-everyone/`. [Online; accessed 22-March-2017].

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, pages 90–94.

Suge Wang, Deyu Li, Xiaolei Song, Yingjie Wei, and Hongxia Li. 2011. A feature selection method based on improved fishers discriminant ratio for text sentiment classification. *Expert Systems with Applications* 38(7):8696–8702.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pages 347–354.

Theresa Wilson, Janyce Wiebe, and Rebecca Hwa. 2004. Just how mad are you? finding strong and weak opinion clauses. In *aaai*. volume 4, pages 761–769.

Huong Nguyen Thi Xuan, Anh Cuong Le, and Le Minh Nguyen. 2012. Linguistic features for subjectivity classification. In *Asian Language Processing (IALP), 2012 International Conference on*. IEEE, pages 17–20.