An Extensive Study of Sentiment Analysis Techniques and its Progression Towards Deep Neural Networks

Shreyansh Sharma[1], Karan Shah[2], Vyom Dave[3], Gopichand G.[4]

[1,2,3]School of Computer Engineering, VIT University, Vellore, Tamil Nadu, India
[4]Assistant Professor, School of Computer Engineering, VIT University, Vellore, Tamil Nadu, India

---***---

Abstract - *In the information age, sentiment analysis has become a must have business tool for SME's and massive industries alike. It has given a new dimension to market analysis. Not so surprisingly, research has increased tenfold and new enhanced techniques are coming in by the day. It is important to study how archaic techniques are getting replaced by state-of-art latest algorithms and get an idea about how the latter benefits more. Traditionally, sentiment analysis was but restricted to basic machine learning algorithms and lexicon based approaches. However, since the turn of the century and the reintroduction of neural networks into the artificial intelligence domain, research has greatly focused on applying these methods to the purpose of sentiment analysis. The paper follows this progression of the evolution of sentiment analysis as a sub domain of natural language processing into an industry feeding tool of its own by employing advanced deep learning models slowly eclipsing the former.*

Key Words: Sentiment Analysis, Machine Learning, Hidden Markov Models, Deep Learning, Convolution Neural Networks, Recurrent Neural Networks

1. INTRODUCTION

In today's world huge amount of data is available that gives reviews or opinions regarding a particular product or service. To analyze this data and take out meaningful information that is useful for many purposes is known as sentiment analysis or opinion mining. Opinion mining is a very helpful tool that when applied correctly can help an organization do wonders in the sense that they get a candid review of their products which can then be used for analyzing the reception of their product. Apart from getting information about a particular product, opinion mining has a lot of other uses such as spam detection, recommendation systems, research purposes, etc. Many social platforms such as blogs and Facebook have a large content of people's opinions which is a good place to start accumulating opinions from. Twitter is a microblogging website that contains posts from people in a short message, this short message makes it easy to analyze the opinions rather than long documents. Such microblogging sites have emoticons, videos and other types of data also embedded into the short messages. Such information is also available in various languages and we can filter the required data using the highly flexible API provided by Twitter.[1] Sentiment analysis was primarily focused on Natural Language Processing in the beginning, but gradually it spread out from computer science to management sciences and social sciences.

From a consumer's point of view, a person wants opinions on many things, such as product review before buying a product, others views about political leaders before voting, about companies before investing, etc.[2] There are many upcoming startups that focus only on sentiment analysis, as this field has very effective real world applications. This research paper aims to elucidate a survey on sentiment analysis using various techniques. Applications of traditional topics such as SVM and Naïve Bayes to many upcoming fields of research including deep learning based sentiment analysis have been elaborately discussed in this paper.

The main aim of this paper is to enlighten the reader on Machine Learning based approaches to sentiment analysis from its inception to the new and upcoming deep learning based sentiment analysis techniques.

2. TRADITIONAL TECHNIQUES

2.1 Machine Learning Based

2.1.1 Using Naïve Bayes Algorithm

One of the most accepted and widely used technique in machine learning classification is Naive Bayes. It comes under the family of 'probabilistic classifiers'. It uses the Bayes' theorem with an assumption of independent relation between the features, hence termed naive'. It uses the chain rule for iterative usage of conditional probability.

The probability of the input with features $x_1, x_2,..., x_n$ to belong to class C_k is defined as:

$$p(C_k, x_1, .., x_n) = p(x_1, .., x_n, C_k)$$

$$p(C_k, x_1, .., x_n) = p(x_1 | x_2, ..., x_n, C_k)\, p(x_2 | x_3, .., x_n, C_k) \ldots p(x_{n-1} | x_n, C_k)\, p(x_n | C_k)\, p(C_k)$$

Used extensively in the fields which require classifications among independent features, like spam classification etc., Naive Bayes is highly scalable, and hence, a very well suited to sentiment analysis. Although it uses the assumption of independent features, which is not always true in the case of sentiment analysis, but in most of the situations, is largely true. The tokens are used as features and the conditional probabilities of the text belonging to a particular class, namely 'positive' or 'negative' is calculated. The larger of the two gets the preference and the text is then classified to that class.

Pablo Gamallo and Marcos Garcia used a variety of techniques to implement Naive Bayes for Sentiment Analysis and compared their efficiency with each other [3]. They based their techniques on the presence of a permutation of the following factors: Unigrams of Lemmas (UL), Valence Shifters (VS, based on negation keywords), polarity lexicon(LEX) and multiwords(MW). Combining different combinations of UL, VS, LEX and MW, they tested the different combinations on constrained (training set) and unconstrained (external datasets). In order to maximize the precision and recall, they used the F-score to judge each technique and found the one which achieves the best efficiency.

Vivek Narayana et al [4] proposed a fast and accurate sentiment classification model using an enhanced version of the Naive Bayes classifier applied on IMDb movie reviews. They added a variety of additional methods on the original Naive Bayes algorithm, like Laplacian Smoothing (to handle new words' probabilities), handling negations, using bigrams and trigrams (n-gram model to intensify positivity or negativity) and feature selection to find a technique that maximizes the accuracy on the test set as well as on an external dataset.

Songbo Tan et al [5] proposed a technique in order to adapt Naive Bayes to Domain Adaptation in order to make the analysis more efficient when it is transferred onto different domains of knowledge. Since, for a trained classifier, it is not easy to be trained on all possible domains, it is proposed that it is trained for all the 'non-domain specific' features. Each text is divided into features which are domain specific(ds), or non-domain specific(nds). The main aim of this technique is to identify not the ds features, but the nds features in a given text and accordingly rate the sentiment using this Adaptive Naive Bayes. Another good approach in order to maximize the efficiency is to go backwards, i.e. judge the accuracy scores, then try to minimize it, and hence identify the problem in the process.

Kang et al [6] proposed a senti-lexicon based improved Naive Bayes algorithms for restaurant reviews. It proposed that in order to minimize the errors in the current Naive Bayes approaches, they tried to minimize the positive sentiment errors by enhancing the positive corpora and to minimize the negative sentiment errors, they enhance the negative corpora. To get the best out of both, they found a middle ground which minimizes both the errors and hence increases the average accuracy of the model. On top of that, they used the n-gram model, and chose their level with repeated tests for higher efficiency. Using this Naive Bayes algorithm, they were able to minimize the errors to up to 3.6% as to when the original one was used.

Although all these techniques aim at maximizing the accuracy and efficiency, in this world of ever increasing data, we not only need to increase the accuracy, but also be able to scale our algorithms to suit Big Data.

Bingwei Liu et al [7] proposed scalable sentiment classification to tackle the problem of such huge amounts of data. They implemented a Naive Bayes Classifier in these large-scale datasets and implemented that on top of the famous MapReduce, a framework used to tackle Big Data, Hadoop being its open source version. They use the contemporary methods along with minor variations to enhance their results and are able to achieve an extremely good accuracy of 82% as the size of the datasets increase. This is vital in the field of Sentiment Analysis to bridge the gap between possible techniques and the practical feasibility in this world of big data.

2.1.2 Using Support Vector Machines (SVMs)

Support Vector Machines, as the name suggests, is a discriminative classifier which uses vectors in a multidimensional graph to form an optimal hyperplane which is able to categorize the data into the different classes. Or in other words, for given p-1 features i.e. there are p dimensional vectors (including the label), SVM aims to find a hyperplane of p-1 dimensions such that it's distance from nearest data point on any of its side is maximized.

Highly used in text and image classifications, handwritten character recognition, biological sciences, SVM has immense use and can be used in a variety of applications. That of course includes sentiment analysis. The enhancements in SVM are highly restricted to the pre-processing as it is a non-probabilistic technique, hence it leaves less scope in terms of algorithmic modifications.

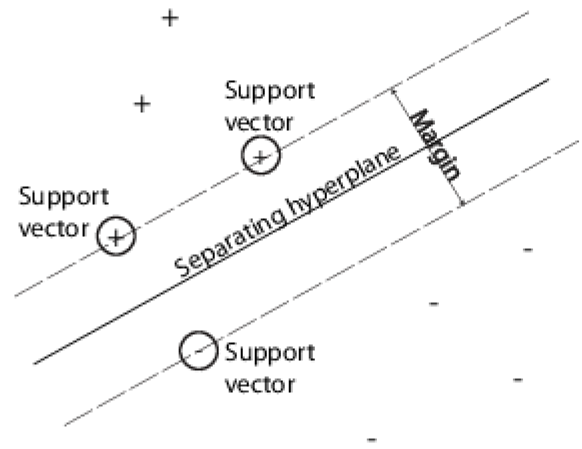

Figure 1: Support Vector Machines for Binary Classification

Nurulhuga Zainuddin et al [8] implemented Sentiment Analysis using Support Vector Machines and, using a variety of tokenization techniques, the n-gram model (unigrams, bigrams, trigrams) and k-folds cross validation, attempted to maximize the accuracy of the created model. The results varied for TFI-DF, Binary Occurrences (BO), Term Occurrences(TO) and was best for TF-IDF.

Shenghua Liu et al [9] proposed an adaptive co-training SVM for sentiment classification of tweets in order to transfer an initial common sentiment classifier into a topic-adaptive classifier. They used simultaneous augmentation of labelled set and expanded the topic-adaptive features in a multiclass SVM model. This process was iterative, resulting in a model which was much more topic-adaptive than its initial iterations.

2.1.3 Using Ensemble Learning

Although the conventional machine learning techniques gave good results, there was a need to try to use multiple techniques at once to maximize the accuracy and efficiency of the tasks. The main agenda of ensemble techniques is to make an amalgam of basic techniques that are individually good but when combined together, they become way more efficient and dependable. It is not necessary that we give equal preference for all the base models, weight based ensemble techniques are also common.

Fersini, Messina, and Pozzi[10] have shown us that we can decide how much a base classifier can contribute in implementing the ensemble technique depending on what part of the sentiment analysis, what base technique is useful and among those which should have a more impact on the ensemble technique as a whole.

As shown by Chun Yang, Xu-Cheng Yin [11], the weighted combination of various base techniques is a very skillful and paramount task which can either make your ensemble a success or a failure.

As evident in Xia, Zong, and Li [12], weighted ensembles are quite effective in implementation of sentiment analysis techniques and provide highly accurate models. Individual models may be quite weak, i.e., each of them may focus only on one aspect of sentiment analysis and when combined together using weights, they may overcome these weaknesses and may provide a pretty convincing and reliable model.

An ensemble technique should have two conditions in order to be classified as good: accuracy and diversity [13]. Diversity is essential because the model should be able to accurately analyze a wide variety of text and in order to achieve that accuracy, the learning algorithm should train on all the different possible variations of sample. To achieve this, data resampling techniques such as bagging and boosting are widely used.

Bagging:

Bagging is one of the simplest ensemble techniques that has very good performance outcomes. Bagging (or Bootstrap aggregation) is achieved by taking random data to form subsets from the entire sampling data with replacement. Each of these subsets is used for different base learner for training purposes. Bagging is usually used where the sampling data is of fixed size. [14][15][16]

Boosting:

Boosting comprises of multiple methods. Boosting creates a number of base learners by training the same learner on different instances of the resampled dataset. [16] Subsequently, the data that is incorrectly classified is then given preference in the subsequent training of the learner. Thus, we can say that the existing learner that has poor performance is trained again to improve the performance.

Generally, basic classifiers such as SVM, NB and ME are used in sentiment analysis as they are quite effective machine learning techniques [17]. Bagging and boosting are compared in several research works, from these, we can conclude that ensemble techniques do not always improve the performance and there are no specific set of rules that can be followed in selection of a particular ensemble technique [18][19].

Catal and Nangir [20] provided a unique ensemble technique that used SVM, Bagging and Naïve Bayes as the base classifiers for sentiment analysis of Turkish problems.

Bagging and Boosting are two of the most popular learning algorithms as proved by their strong results, they try to increase the accuracy by resampling the provided data into various subsets [14].

2.2 Lexicon Based

2.2.1 Using Hidden Markov Models

Markov Chains are nothing but sequences of stochastic or probabilistic events that model any process. They are essentially used for processing text and parts-of-speech tagging before the analysis step. In fact, a hidden markov model approach which is nothing but a statistical form of a regular markov model with unobserved states.

Julian Kupiec[21] describes as early as 1992 how HMMs can be trained with a corpora of untagged text. He rationalized that the tag or the semantic role of the word generally depends on the ones preceding it and those that follow it forming a probabilistic chain. Using this methods, he concludes that millions of dictionary entries can be covered with only a few categories of tags whilst also maintaining a high performance.

Once the part-of-speech tags are available, they can be employed as features to further machine learning algorithms that outputs the sentiment values [22]. In a paper analyzing movie reviews, Oaindrila Das [23] discusses how POS tags alongside the n-gram approach can be used to extract meaning. N-gram approach is nothing but analyzing the N-word window around current word to be analyzed. For instance, a window of size three can be called as a trigram approach. They created a unigram feature matrix based on the term frequencies of the document. A further bigram feature matrix was created to improve the accuracy. The POS tagged output was used to alter the feature matrix generated by giving more weightage to specific terms like the combination of adjectives and nouns; as they quantify the nouns.

Balaji Jagtap[24] uses hidden markov models alongside support vector machines as text classifiers instead of a regular tagger. After the basic preprocessing steps of stemming, tokenizing and removal of stop words, the feature extraction engine calculates the weights bases on the term frequencies which in turn is fed to the HMM. The model is treated as a probabilistic automata consisting of four distinct vector observable forms, namely the initial probabilities, state transition probabilities, set of output symbols and the set of states. The final state of the automaton gives the class of the text. The classifier ran independently alongside an SVM classifier. The combination of both gave the predicted label. They concluded that such hybrid approaches work very well for unstructured data.

There are several other ways wherein modified hidden markov models have been used as classifiers. Liu[25] uses a self-adaptive HMM to classify emotions on Chinese microblogs. They optimize the parameters of the HMM using particle swarm optimization models. They observe that the said self-adaptive HMM outperforms both the Naive Bayes and the Support vector machine algorithms in classifying happiness and sadness based on a given set of tweets. They got an F- Score of 76% on the happiness label and a 65 on sadness. The primary stipulated reason for using a modified HMM as opposed to other mainstream machine learning algorithms is threefold. Firstly, as the training set is in Chinese, segmentation of words will pose a huge challenge as Chinese does not employ apostrophes and spaces similar to English. Secondly, neologisms as similar to English are unknown words and need to be categorized. Thirdly, words are ambiguous and dependent on the context, especially emotional words. Thus, an adaptive HMM is employed that updates model parameters.

Liu explains the procedure as starting with a category based extraction method that focuses more on the predetermined emotion categories rather than extracting all features. The feature vector is not unlike the ones described above and consists the term frequency and the inverse document frequency values. The vector is fed to the HMM. The states of the markov model represent the most apt emotion. There exists a mapping between the feature entered and the HMM state transitions. For each emotion category, a markov model is generated. The feature is passed to all. The final emission probabilities of each model is noted and the class is given to the model with the maximum probability. A stochastic optimizer is built to optimize the state parameters. They analyzed their results using the precision recall and F-Score results. Apart from the scores, they concluded that the classifier could have performed better in instances where there were multiple meanings in a sentence or when the tone was sarcastic.

3. DEEP LEARNING TECHNIQUES

Deep Learning is a part of a broader field of study known as Machine Learning. Deep Learning Neural Networks are inspired by the network of neurons in a human brain. In our brains, many neurons or cells synchronize together and perform so many complex functions, like weighing the different situations and inputs, and accordingly processing it to give out a reaction, an emotion or even just a thought. Deep learning techniques try to imitate that in, of course, a simpler way. A neuron is a fundamental unit of a neural network which takes some input and provides an output according to various conditions. A neural network consisting of a shallow network of such neurons has been around for quite a long time[26]. A Deep Neural Network is such a network that consists of a huge number of layers each consisting of many neurons. As the number of layers and the number of units inside a layer increases, a deep neural network can easily solve tasks having more complexity. Deep learning involves popular techniques such as Multilayer Perceptron networks, Convolutional Neural Networks and Recurrent Neural Networks.

Given their greater understanding as well as its capability to handle much more complex structures, they are very well adapted to tasks like Speech Recognition and NLP. Thus, these deep learning techniques are now also being widely used in the field of Sentiment Analysis, with very promising results.

3.1 Using Convolutional Neural Networks (CNNs):

Convolutional Neural Network (CNN) is a deep neural network which uses the feedforward technique, mostly to analyze visual imagery. This is accomplished by using multiple layers of perceptrons designed in order to require minimal preprocessing. Inspired by our biological design, it attempts to function like a human visual cortex, containing a lot of cells receptive to light in the sub-regions of our visual fields. These are hence called receptive fields.

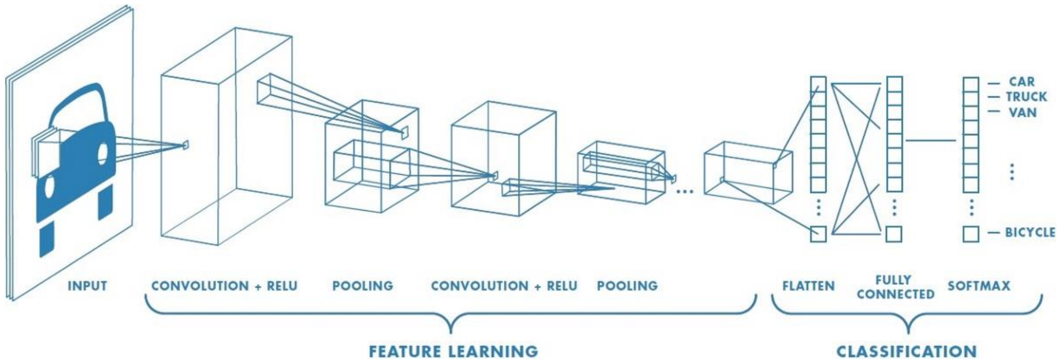

Figure 2: Convolutional Neural Networks with its many layers

The ReLU layer stands for Rectified Linear Units. It uses the non-saturating activation function $f(x) = max(0,x)$ in order to increase the nonlinear properties of the decision function, while not affecting the receptive fields of the convolutional layer.

The Pooling layer is an integral and vital concept of CNN. It is a form of non-linear down-sampling. The pooling can be defined in a variety of ways but max-pooling is the most prevalent of all. The aim of pooling is to partition the input into various non-overlapping sections and then for each section, to output the maximum.

For classification, the softmax function is commonly used for multiclass classification, to represent a categorical distribution in the final layer of the CNN classifier. It is commonly trained under log loss regime, hence giving a non-linear variant of multinomial logistic regression.

These concepts of CNN, even though are mostly used in visual imagery, are very versatile and can be applied to NLP, and sentiment analysis, for better understanding and higher levels of performance.

Aliaksei Severyn and Alessandro Moschitti developed a deep learning system using convolutional neural networks to analyze sentiments of various tweets.[27] They proposed a model using sentence matrix, a convolutional feature map with the trained weights, a pooling function and a softmax function for classification. Using ReLU as the activation function, max pooling, and the softmax function, they used the stochastic gradient descent (SGD) to train the network with the backpropagation algorithm. Updating the learning rate is a technique which can be very helpful in minimizing the processing required and in order to tune it, the Adadelta update rule was used. The obtained model performed well and had a very good accuracy when compared to its non-neural adversaries. CNN techniques offered way better results than other techniques present but was quite saturated on its own. Hence, in order to get even better results, it was combined with various other neural techniques like LSTMs and RNNs.

Jin Wang et al.[28] proposed a dimensional sentiment analysis which used a regional CNN-LSTM model. The LSTM technique, which is an abbreviation of Long Short-Term Memory, are units of a recurrent neural network (RNN). Using regional CNN and

LSTM, they predicted the VA (Valence arousal) of various texts. Max-pooling layer was used to subsample the outputs of the convolutional layer, while LSTM is introduced to capture the long-distance dependency across regions, something other techniques could not make full use of. A linear activation function (linear decoder) is used instead of a softmax classifier to get the sentiment classification. The combination of these two techniques yielded better results as compared to the other models on the sample datasets.

Xingyou Wang et al.[29] developed a neural model combining CNN and RNN for sentiment analysis of short texts. They applied windows of various lengths and weight matrices for convolution. Max pooling was used as the pooling technique, while maintaining the sequential information in sentence context. The model takes in consideration the encoded local features from the CNN model and the long term dependencies from the RNN model. GRU (gated recurrent unit) was also used to make each unit adaptively capture dependencies on different time scales. The results were great and were able to beat most of the other models out there with a very impressive accuracy score.

While we all relate sentiment analysis to texts, Quanzeng You et al[30] proposed a very unique and intriguing technique to analyse the sentiments of images using progressively trained and domain transferred deep networks. CNN, being a big tool in order to visualize, analyze and manipulate visual imagery, was used by the authors to train their model. Using a baseline sentiment algorithm to label Flickr images, they used these images as the training sample to the CNN while employing a progressive strategy (PCNN – Progressive Convolutional Neural Networks) to tune the development. The field of image sentiment analysis is quite a challenging and interesting field, and the results obtained were promising and such techniques could be used for a better understanding and analysis of this field.

3.2 Using Recurrent Neural Networks (RNNs):

Recurrent Neural Networks are a case of Artificial Neural Networks wherein the connections are not only feed-forward, they also form directed cycles. This gives them the ability of having implicit memory and due to this implicit memory RNNs can naturally take time into consideration. RNNs can process complex past signals which are stored and remembered in the memory of the network.[31] RNNs have a number of hidden high dimensional states that serve as memory of the network.[32] The RNNs take advantage of this implicit memory in order to process sequential data. This enables the RNN to be applied to dynamic tasks such as handwriting recognition and speech recognition.

Document level sentiment classification is a very fundamental part of sentimental analysis, and Gated Recurrent Neural Network is one such type of RNN that is used for this task.[33] Meaning of a sentence can be understood by understanding the meaning of its constituents. This model tries to understand sentence representation by using Convolutional Neural Networks or Long Short Term Memory (LSTM). Then, the semantics of the sentences and their relations are then dynamically encoded using the Gated RNN. After understanding the meaning of the constituents, Gated-RNN is exploited extensively to understand the relations between those constituents and how they fit together in order to understand whether a sentence is trying to convey a positive or a negative meaning. As evident in Tang et. al.[33], Convolutional-GRNN is superior than CNN and same applies to LSTM-GRNN. They have performed experiments on four different datasets, and their experiments show that this new novel approach gives state of the art performance on these all datasets and outperform the traditional RNN significantly.

For accomplishing the task of document representation, this method uses two stages. The first stage is to achieve sentence vector representation which is achieved in turn by using word vector representation. A word is represented as a real-valued vector having low dimensions. These vector representations of words are then used to get sentence representation by use of either LSTM or CNN. The second stage involves using these sentence representations which are fed to document composition component which in turn calculates the document representation. The document composition is done by the use of gated recurrent neural network. This method was applied to three restaurant review datasets from Yelp Dataset Challenge (2013, 2014, and 2015) and movie review dataset from IMDB and has given great results in terms of accuracy when compared to the traditional RNN.

Another approach to analyze long texts such as phrases, paragraphs and documents is MT-LSTM (Multi Timescale Long Short Term Memory). LSTM network is a recurrent neural network that is modified to address the problem of blowing up or vanishing.[34] This network remembers important events over a long period of time in which noisy or irrelevant data may also have been trained.[35] Blowing up means that the values of important data become very high due to the noise. Vanishing means that the important values get lost or become very less or insignificant due to the noisy data when the RNN is trained over a long period. LSTM is an extension of Recurrent Neural Network that is highly suitable to model variable length texts.[36] Each LSTM network has a cell that stores valuable information that is updated whenever it encounters valuable material.

In this method, MT-LSM is used to catch the important information by using a variety of timescales. The hidden layers of LSTM are divided into many groups in this process. Each group is activated at different times as per the required time period for storing the data i.e. slow speed groups store long term memories and fast speed groups store the short term memories. The training speed of MT-LSM is approximately 3 times that pf traditional LSTM. One of the important tasks in this method is the selection of number of groups. A long text needs more number of groups than a shorter text. A dynamic method is used for calculating the required number of groups. This method is better than CNN based models in the ways that it requires fewer parameters and achieves approximately the same accuracy, moreover it can also handle a variable length of text. This method gets a boost when the useful information stored in short term memory is transferred to long term memory periodically. The main advantage of this method when compared to many text classification models is that they cannot handle such a wide length of variable texts.

3.3 Using Deep Belief Networks:

As discussed by Geoffrey Hinton[37] in his article, deep belief networks are probabilistic and generative models consisting of several layers of stochastic and latent variables. He says that the latent variables are also called hidden units and consist of binary values. In brief, a deep belief net is a restricted form of a Boltzmann machine and consists of a composition of single learning modules. Hinton[38] also writes that a Boltzmann machine can be simply described as a network of neurons that together make stochastic binary decisions. It has a simple learning algorithm that spots irregularities in data. Thus, a restricted Boltzmann machine as used in a deep belief network, consists of just one layer of feature detectors or neurons making it faster to train. The belief net probabilistically reconstructs its input vectors and forms a feature detector which can be used to perform classification. It can even contain multiple layers of restricted Boltzmann machines. In the case of sentiment analysis, it can be used to classify document polarities or even sentences.

Zhou et. al.[39] in their paper suggest a novel deep learning structure called as weakly shared deep neural networks or WSDNNs that is adapted on a simple deep belief net. It builds a weakly shared layer between two languages, one which is label rich and the other that is not and tries to assign a polarity in sentiment to the text in the latter. They use the Prettenhofer and Stein dataset that consists of corpora in languages like English, German, French and Japanese. This method is very widely used even in the industry, most notably in the product reviews on Amazon.

A hybrid approach was notably developed by Chen et. al.[40] wherein they create a hybrid deep belief network that works as a semi supervised learning algorithm. They assert that a lot of sentiment analysis data is semi supervised as a lot of it is unlabeled. Hence they generate a novel idea to work with. The architecture is threefold. In the first step, several hidden layers of a restricted Boltzmann machines are created that can reduce the dimensions of review data. It must be noted that most training data used for sentiment analysis is obtained from user reviews. In the second stage, they hidden layers using convoluted hidden layers, that can extract features from the data. Finally, a gradient descent based approach is used to alongside a loss function to facilitate learning. They test the algorithm on five sentiment datasets. The first is the famous MOV corpus that contains movie reviews while the other are obtained from multiple domains. Each of them are of size 2000 with equal positive and negative sentiments. For the purposes of this experiment, they used 2 regular hidden layers and one convoluted hidden layer. The most important aspect was to compare the performance of the hybrid deep belief network algorithm with other algorithms like the semi-supervised spectral learning, SVMs, regular belief networks etc. The results were highly promising with the HDBN approach showing the highest accuracy in four of the five datasets used while coming second by a measure of just 2 percent in one. They concluded by observing that the reason for high performance of the HDBN in sentiment analysis can be credited to the fact that it can use a high amount of unlabeled data alongside labeled data in one model and thus it outperformed other well-known supervised learning algorithms.

4. CONCLUSIONS

Sentiment Analysis is an ever growing and rapidly evolving field in the technological world. With a plethora of applications especially in the business markets that are so widely dependent on online activities of billions of people, it is a field that will always develop with the onset of every new technology available. Given its complex structure, it is necessary to carefully understand the techniques and adapt it towards sentiment analysis accordingly.

After carefully analyzing the popular and less known algorithms to obtain the sentiments, we see how the methods evolved over the years and is at its epitome with deep learning. We can, for now, safely arrive at the conclusion that the deep learning techniques provide us way more accurate results as compared to its other counterparts.

REFERENCES

[1] Pak, A., & Paroubek, P., "Twitter as a corpus for sentiment analysis and opinion mining", 2010, May, LREc (Vol. 10, No. 2010, pp. 1320-1326).

[2] Liu, B. (2012). "Sentiment analysis and opinion mining", 2012, Synthesis lectures on human language technologies, 5(1), 1-167.

[3] Pablo Gamallo, Marcos Garcia, "Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets", 2014, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)

[4] Vivek Narayanan, Ishan Arora, Arjun Bhatia, "Fast and accurate sentiment classification using an enhanced Naive Bayes model", Intelligent Data Engineering and Automated Learning IDEAL 2013 Lecture Notes in Computer Science Volume 8206

[5] Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu, "Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis", in Boughanem M., Berrut C., Mothe J., Soule- Dupuy C. (eds) Advances in Information Retrieval. ECIR 2009. Lecture Notes in Computer Science, vol 5478. Springer, Berlin, Heidelberg
[6] Kang, Hanhoon, Joon Yoo, Seong, and Han "Senti-lexicon and improved Native Bayes algorithms for sentiment analysis of restaurant reviews" Expert Systems with Applications, Dongil 2012
[7] Liu, Bingwei, Erik Blasch, Yu Chen, Dan Shen and Genshe Chen, "Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier." 2013 IEEE International Conference on Big Data (2013): 99-104.
[8] Nurulhuda Zainuddin and Ali Selamat, "Sentiment Analysis Using Support Vector Machine", IEEE 2014 International Conference on Computer, Communication, and Control Technology
[9] Shenghua Liu, Fuxin Li, Fangtao Li, Xueqi Cheng and Huawei Shen, "Adaptive Co-Training SVM for Sentiment Classification on Tweets", In Proceedings of the 22nd ACM international conference on Conference on Information and Knowledge Management (CIKM'13), 2013
[10] Fersini, E., Messina, E., and Pozzi, F. A., "Sentiment analysis: Bayesian ensemble learning", 2014, Decision support systems, 68, 26-38.
[11] Yang, C., Yin, X. C.,and Hao, H. W. "Diversity-based ensemble with sample weight learning" in Pattern Recognition (ICPR), 2014 22nd International Conference on (pp. 1236-1241). IEEE.
[12] Xia, R., Zong, C., and Li, S. "Ensemble of feature sets and classification algorithms for sentiment classification", 2011, Information Sciences, 181(6), 1138-1152.
[13] Windeatt, T. and Ardeshir, G. "Decision tree simplification for classifier ensembles", 2004, International Journal of Pattern Recognition and Artificial Intelligence, 18(05), 749-776.
[14] Oza, N. C. "Online bagging and boosting", 2005
[15] Maclin, R. and Opitz, D.,"An empirical evaluation of bagging and boosting", 1997, AAAI/IAAI, 1997, 546-551.
[16] Wang, G., Sun, J., Ma, J., Xu, K., and Gu, J. "Sentiment classification: The contribution of ensemble learning", 2014, Decision support systems, 57, 77-93.
[17] Pang, B., Lee, L., and Vaithyanathan, S. "Thumbs up?: sentiment classification using machine learning techniques", July 2002, in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.
[18] Whitehead, M., and Yaeger, L. "Sentiment mining using ensemble classification models", 2010, In Innovations and advances in computer sciences and engineering (pp. 509-514). Springer, Dordrecht.
[19] Lin, Y., Wang, X., Li, Y. and Zhou, A., "Integrating the optimal classifier set for sentiment analysis", 2015, Social Network Analysis and Mining, 5(1), 50.
[20] Catal, C., and Nangir, M. "A sentiment classification model based on multiple classifiers", 2017, Applied Soft Computing, 50, 135-141.
[21] Julian Kupiec "Robust part-of-speech tagging using a hidden Markov model", Computer Speech and Language 1992
[22] Bo Pang "Thumbs up? Sentiment classification using machine learning techniques" Conference on Empirical Methods in Natural Language Processing 2002
[23] Oaindrila Das, Rakesh Chandra Balabantaray, "Sentiment Analysis of Movie Reviews using POS tags and Term Frequencies" International Journal of Computer Applications 2014
[24] Balaji Jagtap, Virendrakumar Dhotre "SVM and HMM Based Hybrid Approach of Sentiment Analysis for Teacher Feedback Assessment" International Journal of Emerging Trends & Technology in Computer Science 2014
[25] Li Liu, Dashi Luo, Ming Liu, Jun Zhong, Ye Wei, Letian Sun, "A Self-Adaptive Hidden Markov Model for Emotion Classification in Chinese Microblogs", Mathematical Problems in Engineering 2015
[26] Schmidhuber J., "Deep learning in neural networks: An overview", 2015, Neural networks, 61, 85- 117.
[27] Aliaksei Severyn and Alessandro Moschitti., "Twitter Sentiment Analysis with Deep Convolutional Neural Networks", 2015, 38th International ACM SIGIR Conference 959-962
[28] Jin Wang, Liang-Chih Yu, K. Robert Lai and Xuejie Zhang, "Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model", 2016, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 225-230.
[29] Xingyou Wang, Weijie Jiang and Zhiyong Luo, "Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts", 2016, COLING.
[30] Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang "Robust image sentiment analysis using progressively trained and domain transferred deep networks", 2015, In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15). AAAI Press 381-388.
[31] Salehinejad, H., Baarbe, J., Sankar, S., Barfett, J., Colak, E., and Valaee, S., "Recent Advances in Recurrent Neural Networks", 2017, arXiv preprint arXiv:1801.01078.
[32] Mikolov, T., Joulin, A., Chopra, S., Mathieu, M., and Ranzato, M. A. "Learning longer memory in recurrent neural networks", 2014, arXiv preprint arXiv:1412.7753.
[33] Tang, D., Qin, B., and Liu T., "Document modeling with gated recurrent neural network for sentiment classification". In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 1422-1432).

[34] Hochreiter, S., and Schmidhuber, J. "Long short-term memory", 1997, Neural computation, 9(8), 1735-1780.
[35] Graves, A. "Long short-term memory", 2012, In Supervised Sequence Labelling with Recurrent Neural Networks (pp. 37-45). Springer, Berlin, Heidelberg.
[36] Liu, P., Qiu, X., Chen, X., Wu, S., and Huang, X. "Multi-timescale long short-term memory neural network for modelling sentences and documents", In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 2326-2335).
[37] Geoffrey E. Hinton, "Deep belief networks", Scholarpedia, 2009, 4(5):5947.
[38] Geoffrey E. Hinton, "Boltzmann machine", Scholarpedia, 2007, 2(5):1668.
[39] Guangyou Zhou, Zhao Zeng, Jimmy Xiangji Huang , and Tingting He, "Transfer Learning for Cross-Lingual Sentiment Classification with Weakly Shared Deep Neural Networks", Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, 2016
[40] Shusen Zhou, Qingcai Chen, Xiaolong Wang, Xiaoling Li, "Hybrid Deep Belief Networks for Semi-Supervised Sentiment Classification" Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers

Figure 1: Support Vector Machines for Binary Classification, Mathworks: https://in.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html
Figure 2: Convolutional Neural Networks, Mathworks: https://www.mathworks.com/discovery/convolutional-neural-network.html