University of Ibadan
Journal of Science and Logics in ICT Research

Semantic Sentiment Analysis Based on Probabilistic Graphical Models and Recurrent Neural Networks

[1]Osisiogu, U. O., and [2]Odumuyiwa, V.T.
[1]*Department of Computer Science, African University of Science and Technology, Abuja, Nigeria*
[2]*Department of Computer Sciences, University of Lagos, Akoka, Nigeria*
uosisiogu@aust.edu.ng, vodumuyiwa@unilag.edu.ng

Abstract
Sentiment Analysis is the task of determining the sentiment polarity expressed in textual documents. This can be achieved by using lexical and semantic methods. The purpose of this study is to investigate the use of semantics to perform sentiment analysis based on probabilistic graphical models and recurrent neural networks (RNN). In our empirical evaluation, the classification performance of the graphical models was compared with some traditional machine learning classifiers and a recurrent neural network. The datasets used for the experiments were IMDB movie, Amazon Consumer Product reviews, and Twitter Review datasets. Obtained results from empirical study show that semantic representation of textual documents using word embeddings in conjunction with long-short term memory (a RNN family) for classification produces better result in determining the polarity expressed in texts.

Keywords: Semantic Sentiment Analysis, Recurrent Neural Networks, Probabilistic Graphical Models, Natural Language Processing.

1. Introduction

Sentiment analysis is the subject of natural language processing technique whose main aim is to perform the task of classifying, extracting and detecting attitudes, sentiments, and opinions of the different aspects or topics of an entity or product expressed in textual form. The usefulness of sentiment analysis includes but is not limited to determining the level of consumer satisfaction [1], analysing political movements [2], performing market intelligence [3], measuring and improving brand reputation [4], box office prediction [5], and many others [6], [7].

Access to people's opinions, sentiments and evaluations have increased in general and in a wide variety of fields such as e-commerce [8], tourism [9], and social networks [10]. Consumers now read product reviews by previous customers. In addition, improvement of products and services carried out by service providers is enhanced through feedback obtained from customers through channels that employ textual data.

Despite the stated usefulness and advantages that come with sentiment analysis, automatically determining sentiments expressed in textual documents is faced with a lot of challenges. These challenges include: the usage of sarcastic statements especially in social network platforms like Twitter; the possibility of words possessing different meanings, for instance, a word can bear positive meanings in some contexts, and negative in another; people also express their opinions in varied ways so a small change in the syntax of the message communicated can mean something different in the implied opinion. Also, some of the opinions expressed cannot be categorized as a particular type of sentiment, because sometimes they may appear to be subjective and also appear neutral in another perspective. Issues like these could raise questions like "at what point can we classify a statement as being neutral or positive or negative?" The aforementioned shows how challenging sentiment analysis can be, even for humans.

This paper seeks to verify if the semantic representation of data can further inform the classification process of an algorithm.

Osisiogu, U. O., and Odumuyiwa, V. T. (2022). Semantic Sentiment Analysis Based on Probabilistic Graphical Models and Recurrent Neural Networks, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 7 No. 2, pp. 33 - 44

Probabilistic graphical models and Recurrent Neural Network (RNN) were used to perform semantic sentiment analysis of textual data. The focus on probabilistic graphical models (PGMs) are emphasized because of their ability to model dependencies between events. Performance evaluation of graphical models, traditional machine learning algorithms and RNN in sentiment analysis tasks on some benchmark datasets was also carried out.

Section 2 of this paper presents a comprehensive review of the previous and related works that have used probabilistic graphical models, traditional machine learning algorithms and neural network for text analytics. In Section 3, the methodologies and the approach used for the proposed investigation were presented. Section 4 presents and discusses the results obtained from the experiments carried out. Afterwards, conclusion is presented in section 5.

2. Literature Review

This section presents the existing works on sentiment analysis and briefly discusses the classification methods used for experimental purpose in section 3.

2.1 Related Works

Using sentiment lexicon alone to carry out sentiment analysis has been in vogue for quite some time, however, the lexicon-based approach to sentiment analysis fails to capture the meaning of words in the context the word is used. It takes polarity count of words in a document and assigns the highest polarity to the document without considering the meaning of the words as it relates with the document. Thus, the need for semantic sentiment analysis which involves getting the computer to learn to reason like human when analyzing any document.

Wan's work [11] and an article by Al-Smadi *et al.* [12] show that one of the recurrent use of Bayesian Networks (BN) is classification as they are directly used as sentiment classifiers in these works. They obtained competitive results and in some cases higher when compared with other approaches. In another work proposed by Chen *et al.* [13], a parallel algorithm for the structure learning of large-scale text datasets for Bayesian networks was created. With the application of a MapReduce cluster, dependencies between words were captured. This approach allows for obtaining a vocabulary for extracting sentiments. Experiments were carried out using a blog's dataset; this work points out that features can be extracted despite fewer predictor variables.

A two-stage Markov Blanket Classifier was proposed by Airoldi, Bai and Padman [14] to perform extraction of sentiments from unstructured text, such as film reviews, using BNs. In their approach, a Tabu Search algorithm [15] was used to prune the resulting network to obtain more accurate classification results. Although this helps to prevent overfitting, their work does not efficiently exploit dependencies among sentiments. In contrast, Orimaye [16] proposed an improvement for the Bayesian Network classification model that fully exploits sentiment dependencies by including sentiment-dependent penalties for scoring functions of Bayesian Networks (e.g. K2, Entropy, MDL, and BDeu). This proposed modification derives the dependency structure of sentiments using conditional mutual information between each pair of variables in the dataset. In Orimaye *et al.*, [17] the knowledge contained in SentiWordNet was evaluated. The experimental results obtained showed that this sentiment-dependent model could improve the classification accuracy in some domains.

A hierarchical approach for the modelling of simple and complex emotions in texts is proposed by Ren and Kang [18]. Many documents contain complex human emotions. Such emotions are a mixture of simple emotions which may not be easily modelled using traditional machine learning techniques (e.g., Naïve Bayes, and Support Vector Machine). The traditional machine learning algorithms were able to model texts with simple emotions while the hierarchical methods were more suitable for modelling documents with complex emotions. The analysis performed in this work also points out that there is a relationship between the topics of documents and the emotions contained in them.

Lane et al. [19] addressed issues facing most sentiment analysis tasks such as choosing the right model, feature extraction and dealing with unbalanced data. Although the main task is classification, they took into consideration two different techniques. First, the classification subjectivity and second, the polarity determination. Several techniques for extracting features were evaluated, as dealing with unbalanced data was considered before training. It turns out that the Bayesian Network model tested showed a decrease in their performance when applying data balancing techniques. This behaviour was different from that of the other classifiers.

Kang *et al.* [20] focused on the sequences of words to address some of the issues faced with the use of lexicons when performing sentiment analysis. They proposed the use of a model that focusses on word orders without the need for extracting sentiment lexicons. To achieve this an ensemble of text-based Hidden Markov Model (HMM) is proposed. This model employed the boosting and clustering of words produced by latent semantic analysis. After the input data has been labelled and words in the textual data have been clustered, the ensemble is used to create a classifier.

2.2 Classifiers

2.2.1 Bayesian Network Classifiers

Sentiment analysis problems can be approached through a Probabilistic Graphical Model known as a Bayesian Network; Bayesian networks are modelling techniques that allow for the description of dependency relationships between different variables by the application of a directed graph structure that encodes conditional probability distributions [21]. By storing expert knowledge in the structure of these models, Bayesian Probabilistic models can perform or support classification tasks [22]. Following the context of modelling and machine learning problems, Bayesian networks are normally used to find relationships among a large number of words. Thus, BNs provide an adequate tool used to model these relationships. BNs consists of a directed acyclic graph where each node represents a random variable and the edges between the nodes represent an influence relationship. Conditional probability distributions are typically used to model these influences.

To define conditional probability distributions a table known as Conditional Probability Table (CPT) is given. To build a classifier using Bayesian Network, it is required that the structure of the network is first learned along with their respective CPTs. Furthermore, the fundamental concept of CPTs can be extended to the continuous case in which the variables can base on the other laws of probability such as Gaussian Distribution or solved by applying discretization [23],[24],[25]. Although inference in any Bayesian Network is an NP-hard problem[26], there are efficient alternatives that exploit conditional independence for some types of networks [27]. Also, one of the benefits of Bayesian Networks in their ability to directly handle incomplete datasets if one of their entries are missing.

2.2.2 Logistic Regression

One of the foremost methods of text classification algorithms is known as Logistic Regression (LR). This algorithm was introduced and developed by a statistician known as David Cox [28]. LR is a linear classifier with a decision boundary defined by $\theta^T x = 0$ and it predicts probabilities rather than classes [29][30]. To define a class, it takes the maximum value of the predicted probability of the respective class. However, there are certain limitations to this algorithm; LR classifiers work well for predicting categorical outcomes. To ensure optimal performance, the prediction requires that each data point be independent identically distributed (iid) to perform best. These data points attempt to predict the outcomes based on a set of independent variables[31].

2.2.3 Naïve Bayes Classifier (NBC)

Naïve Bayes Classification has been widely used for text classification tasks that involve document categorization tasks [32]. The Naïve Bayes method is based on Bayes theorem, formulated by Thomas Bayes [33]. Information retrieval systems have widely adopted this algorithm [34]. This technique is a generative model – a traditional method of text categorization. In this project, we apply the Naïve Bayes classifier on textual data that has its feature extracted by the Term Frequency-Inverse Document Frequency (TF-IDF) approach. One peculiar limitation of the NB classifiers is its inability to work on unbalanced classes.

2.2.4 Support Vector Machine (SVM)

Vapnik and Chervonenkis [35] developed the original version of SVM in 1963. Although the SVM was designed for binary classifications, many researchers work on multi-class problems using this technique.

2.2.5 Decision Trees (DT)

Decision Tree classifiers have been successfully used in varied areas of classification. It was introduced as a classification tool by Magerman [36] and inductions developed by Quinlan[37]. This technique employs a hierarchical composition of the data space. The main idea behind this algorithm is found upon the creation of a tree based on the attribute for categorized data points. A major challenge in the implementation of a decision tree is in the assignment of attributes to the parents' level or the child level.

2.2.6 Random Forest (RF)

One of the ensemble learning methods that is mainly used in text classification tasks is known as Random Forests or Random Decision Forests technique. This technique was introduced by Kam Ho in 1995 [38][39]. The decision trees generate random decision trees that are trained and predictions are assigned by voting. Some of the limitations of Random Forest remain that they are quite slow to create predictions once trained. However, they possess a better speed of convergence when compared with other machine learning algorithms. To achieve faster prediction results the number of trees in the forest must be reduced, this can result in lesser time complexity in the prediction step.

2.2.7 Long Short-Term Memory (LSTM)

Neural Networks are designed to learn a multi-connection of layers that every single layer only receives the connection from the previous and provides connections only to the next layer in a hidden part. An important variation of this that has been utilized by several researchers for text mining and classification tasks is the recurrent neural network (RNN) [40]. The RNN assigns more weights to the previous data points of a sequence. Thus, this feature makes the RNN a powerful approach to sequential data, text, and strings. LSTM is a special type of RNN that addresses the problem of vanishing gradients by preserving long term dependencies more effectively when compared to the basic RNN.

Hochreiter & Schmidhuber introduced the LSTM [41], ever since this architecture has been augmented by many research scientists. LSTM possess a chain-like structure similar to RNN, LSTM utilizes multiple gates to carefully regulate the degree of information that is allowed into each node state. A form of bias can be introduced into RNNs when later words are more influential than earlier ones. This, however, can be resolved with the deployment of max-pooling areas.

3. Methodology

The scope of this work lies within the investigation of semantic representation and semantic feature extractions from textual data for sentiment analysis. The import of Probabilistic Graphical Models (PGMs) especially Bayes Network, some traditional machine learning algorithms and a recurrent neural network model in capturing the semantics of textual documents for sentiment analysis was evaluated in an experimental process.

3.1 Datasets

The datasets used in the empirical research were IMDB movie reviews, Amazon Product reviews, and Twitter datasets. These datasets are the most common datasets used in the literature as benchmark datasets for sentiment analysis.

3.1.1 IMDB Dataset

This is a dataset for binary sentiment classification containing 25,000 highly polar movie reviews for training, and 25,000 for testing [42].

3.1.2 Amazon Product Review

This dataset is a subset of the main dataset that contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. This dataset includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs). For this paper, only about 28,000 records from the dataset were used and some resampling were performed where necessary.

3.1.3 Twitter Datasets

This dataset consists of 4,242 tweets manually labelled with their polarity.

3.2 Workflow

Figure 1 shows the workflow for this research, and does not differ from most conventional research methodology in NLP thus the reason for its adaptation.

First, data preprocessing was done to generate representations of the text documents. The preprocessing method differs for each classifier: while in some cases the texts are represented as feature vectors (TF-IDF), in some other cases

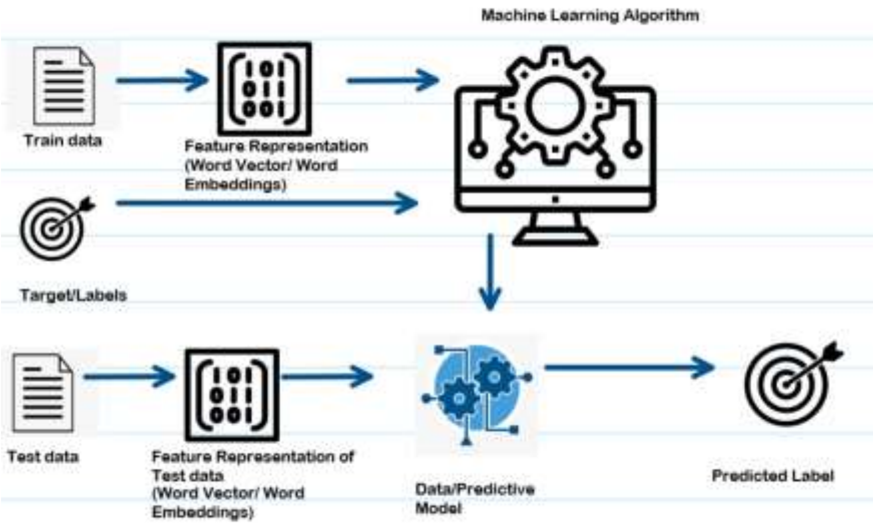

Figure1: Research Workflow

they are represented as Word Embeddings. The different representations are fed into their respective algorithms to train the models. After the model has been trained, new text documents (test set) are fed into the trained model and of course, are also represented in the same way as the test dataset. With this, the models to classify the new record and the predicted class is checked against the actual class of the test dataset to measure the performance of the model. Table 1 shows the algorithms and the textual representation used for them.

3.3 Classification Methods and Algorithms

The Bayesian Network used in this experiment were obtained from Weka [43]; the implementation of the traditional machine learning classifiers (Logistic Regression, Support Vector Machine, Naïve Bayes, Decision Trees, Random Forest) used was obtained from the SciKit Learn API [44] built for machine learning. TensorFlow [45] was used for the neural network implementation.

3.3.1 Graphical Models

In this experiment, the Bayesian Network (BN) as implemented in WEKA was used. Emphasis was on the use of different scoring functions and search algorithms on the BN. The scoring functions used are Bayes, BDeu, MDL, Entropy and AIC. The search algorithms used to search the space are: K2 [46], Hill Climbing [47], Repeated Hill Climber, LAGD Hill Climber and Tabu Search [48].

The datasets were prepared according to the WEKA's ARFF format by concatenating the negative and positive reviews for each dataset and created a string data file in the ARFF format. The string data file was then preprocessed using the *weka.filters.unsupervised.attribute.StringToWordVector* package.

This package converted the string data file to a TFIDF data file in the ARFF format to produce a numerical representation of the text variables that are supported by the *Bayes* package. This representation still maintains the dependency relationship between words (variables) as in the original string format. Table 2 shows the number of attributes used. This was carefully selected after testing a range of attributes, the number of attributes that resulted in having the best performance was then selected.

Table 1: Outline of experiments carried out.

Method	Algorithm	Textual Representation
Graphical Models	Bayesian Network Classifier	TF-IDF (Word Vectors - Sparse Vector Representation)
Non-Semantic (using traditional machine learning algorithms)	Logistic Regression Support Vector Machine Naïve Bayes Decision Trees Random Forest	TF-IDF (Sparse Vector Representation)
Semantic Representation	Long Short-Term Memory	Word Embeddings (GloVe, Word2Vec) (Dense Vector Representation)

Table 2: Distribution of prepared datasets for Bayesian Network used in WEKA.

Dataset	Instances	Negative/Positive	Attributes
IMDB	50000	25000/25000	5000
Amazon	28332	8435/19897	2500
Twitter	4438	2218/2218	65

3.3.2 Machine Learning Classifiers

Some machine learning classifiers were also used to detect the sentiment polarity of the records in the three datasets. The classifiers used are Naïve Bayes, Support Vector Machine (SVM), Logistic Regression (LR), Decision Tree (DT) and Random Forest (RF). For each dataset, each of the reviews was pre-processed by removing punctuations, converting URLs to string "URL", removing numbers and symbols to obtain alphanumeric data, coercing string to lowercase and using the s*klearn.feature_extraction* package to apply the TF-IDF vectorizer to generate sparse vector representation.

After the preprocessing the dataset was split into 80% for training and 20% for testing to evaluate the performance of the models.

3.3.3 Recurrent Neural Networks – (Long- and Short-Term Memory)

Recurrent Neural Networks with LSTM layers were implemented in this experiment to demonstrate the use of semantics in sentiment analysis. To implement the textual semantic representations of words, word embeddings were used. Dense vector representations were used to train our semantic models. In this experiment we utilized two types of word embeddings namely: Word2Vec [49] and Global Vectors (Glove).

Figure 2 shows the architecture that was implemented in building the neural network classifier using the TensorFlow deep learning framework. To improve the performance of the neural network hyper-parameter optimizations were carried out. Table 3 shows the parameters that were used. For the Twitter dataset, we used 20 epochs, for the Amazon product reviews 25 epochs were used, while the IMDB dataset 30 epochs were used. This batch sizes used for the different datasets also follows the aforementioned order.

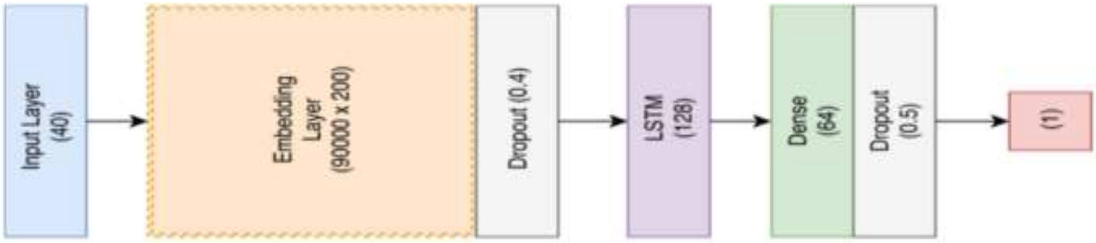

Figure 2: LSTM architecture used

Table 3: Hyper-parameters of the LSTM implemented

Hyperparameter	**Hyperparameter implemented**	**Remarks**
Optimizer	Adam Optimizer	Gave us the highest accuracy
Loss Function	Binary Cross-Entropy loss	Most suitable for Binary classification tasks
Epochs	20, 25, 30	Varies for the dataset used
Batch Size	50, 100, 150	Varies for the dataset used

4.0 Results and Discussion

4.1. Results

Results obtained from the experiments based on the different methods presented in Table 1 are shown in Tables 4-6.

Table 4: Precision, Recall and F1 score of the respective Bayes classifier using different search algorithms

	IMDB			**Amazon**			**Twitter**		
Search algorithms	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
K2	0.857	0.857	0.857	0.730	0.747	0.738	0.683	0.644	0.663
Hill Climber	0.857	0.857	0.857	0.730	0.747	0.738	0.683	0.644	0.663
LAGD Hill Climber	0.857	0.857	0.857	0.730	0.747	0.738	0.683	0.644	0.663
Repeated Hill Climber	0.857	0.857	0.857	0.730	0.747	0.738	0.683	0.644	0.663
Tabu Search	0.857	0.857	0.857	0.730	0.747	0.738	0.683	0.644	0.663
TAN	0.858	0.857	0.857	0.735	0.750	0.742	0.685	0.646	0.665

Table 5: Results obtained from the machine learning classifiers on the three datasets

	IMDB			Amazon (with resampling)			Twitter		
Classifier	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Naïve Bayes (NB)	0.8972	0.8969	0.8969	0.9315	0.8464	0.8775	0.6845	0.6847	0.6844
Support Vector Machine (SVM)	0.9001	0.8999	0.8999	0.9217	0.8872	0.9015	0.6613	0.6610	0.6611
Logistic Regression (LR)	0.8609	0.8596	0.8595	0.9300	0.8722	0.8942	0.6892	0.6881	0.6881
Decision Trees (DT)	0.7168	0.7168	0.7168	0.9039	0.8783	0.8899	0.6270	0.6239	0.6234
Random Forest (RF)	0.7465	0.7414	0.7402	0.9068	0.9298	0.9133	0.6316	0.6227	0.6199

Table 6: Summary of the results of the LSTM implemented

Dataset	**Classifier**	**Feature Representation**	**Accuracy**
IMDB Movie Review	LSTM	Word2Vec	88.64%
		GloVe	89.12%
Amazon Product Consumer Review	LSTM	Word2Vec	98.18%
		GloVe	97.44%
Twitter Dataset	LSTM	Word2Vec	89.82%
		GloVe	91.20%

Table 7: Summary of the accuracy of the classifiers used in experimentation.

Classifier	**Dataset**		
	IMDB Movie Reviews (%)	**Amazon Product Reviews (%)**	**Twitter dataset (%)**
Bayesian Network (BN)	85.80	74.20	66.50
Logistic Regression (LR)	89.69	89.42	68.81
Support Vector Machine (SVM)	89.99	90.15	66.11
Naïve Bayes (NB)	85.95	88.72	68.44
Decision Trees (DT)	71.68	88.99	62.34
Random Forest (RF)	74.02	91.33	61.99
RNN (LSTM)	89.12	98.18	91.20

4.2 Discussion of Results

Table 4 shows the summary of performance of the Bayesian Network classifier when applied to the IMDB movie review, Amazon Product review and Twitter dataset respectively. For each search algorithm, different scoring algorithms were used and the results obtained were the same except for the Tree Augmented Naive (TAN) Bayes algorithm which consistently obtained a slightly better result. We observed no difference in performance for the different scoring functions used. The IMDB movie review has the highest amount of accuracy mainly because it has the largest amount of dataset when compared to the other datasets. For the Amazon product review dataset, the precision and recall of the traditional machine learning algorithms was affected because of the imbalanced data. The results obtained show that the imbalanced nature of the dataset causes a little effect in the resulting precision and recall figures.

Table 5 presents the result obtained on experimenting with the five machine learning algorithms with the assumption that such algorithms do not emphasize on the semantics of the text document. The goal was to compare their performance with those of the graphical method and the LSTM with the assumption that the duo (graphical method and LSTM) captures the semantics between texts in a document. Table 5 shows that the performance of the machine-learning algorithms in the classification tasks depends to an extent on the number of datasets available. The machine learning algorithms on the IMDB datasets achieved greater prediction accuracy than the other datasets. Also, SVM performs better than the other traditional methods. The results obtained from the operation of the machine learning algorithms for the Amazon Product Review datasets show a significant difference between the Micro-average and Macro average, this is as a result of the unbalanced data sets. Although a resampling process (up sampling) was carried out, the classification algorithms made better predictions with the larger class (positive reviews). The micro-average results of the Random Forest Classifier outperform that of the SVM classifier although the latter's macro-average significantly outperforms the former. The results obtained from the operation of the classifiers on the Twitter Datasets further supports the strong correlation between classification accuracy and the number of data samples. However, the classifier with the best results is the Logistic Regression Classifier and following that is the Naïve Bayes algorithm. This shows promising results as the size of the Twitter datasets are relatively small. This discovery calls for further investigation in developing methods that can harness the strengths of various algorithms to achieve optimal accuracy.

In Table 6, it can be seen that the LSTM when applied to GloVe, produced the best results. It is worthy of note that one of the reasons why such results were produced is that GloVe tends to encode a better level of semantics when compared to Word2Vec embeddings. In this experiment, an external word embedding was used. The GloVe with 6 billion tokens and a dimension of 100 was utilized to carry out the sentiment analysis task. This embedding was trained using the Wikipedia corpus. With these embeddings, the neural network was trained and used to make classifications based on the semantic encodings from the word embeddings. LSTM was able to capture to some extent the semantic and syntactic features of the textual sentiment. This is one of the reasons why it consistently outperforms other classifiers in most cases. However, certain limitations still occur when using this classifier. It performs badly when the datasets available are little. It also suffers from a lack of interpretability as we cannot certainly ascertain how the classification is being performed by the algorithm. Unlike the SVM which does not require a lot of hyper-parameters tuning a lot of tuning has to be made to get the best out of the application of this technique.

The summary of results in Table 7 show that Graphical Models like Bayesian Networks also give reasonable results when being used to perform sentiment analysis tasks. The bar chart in Figure 3 helps to visualize these results. The SVM, as seen in Table 7, consistently provides one of the best results because it can produce an efficient separation of classes when features are well represented using vectors. Effective and efficient text representation for Graphical models that can encode semantics can somewhat improve the way they perform text classification tasks. Further study on how the inclusion of semantics will not only be done in the scoring or learning algorithm but also on the text feature representation may improve the performance of graphical models for the sentiment analysis task.

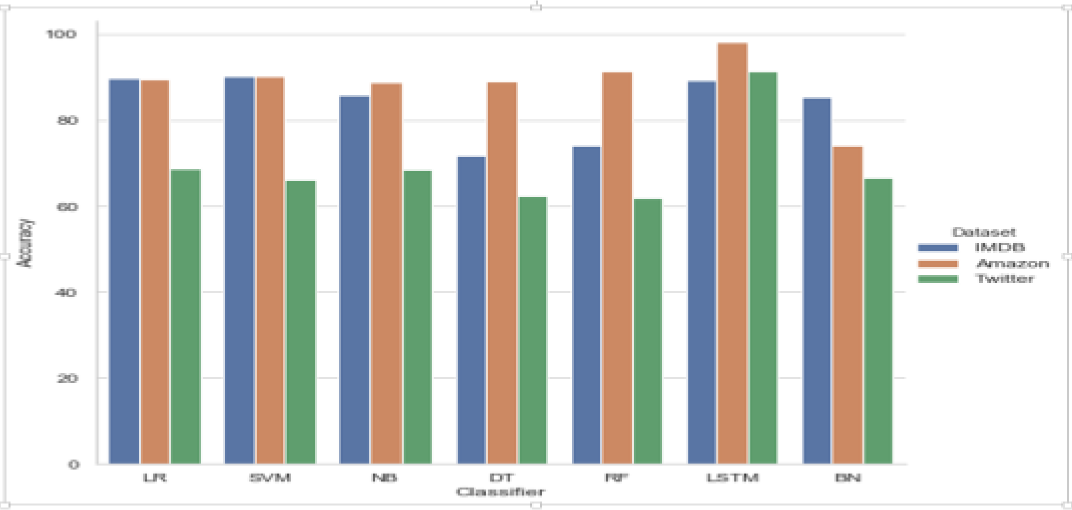

Figure 3: Performance accuracy of the classifiers on the three datasets

5. Conclusion

In this work, sentiment analysis, its usefulness and applications were pointed out. The need for the improvement of this task led to several investigations of how well various machine learning classifiers can be used to carry out this task. As these machine learning classifiers show comparable results, this work focusses more on how the semantics of texts and dependencies between texts in a document can be taken into consideration in performing the sentiment analysis task. This work pointed out how Graphical models and neural networks encode semantics of texts in their various methods. Obtained results from empirical study show that semantic representation of textual documents using word embeddings in conjunction with long-short term memory (a RNN family) for classification produces better result in determining the polarity expressed in texts.

References

[1] Ren, F., & Quan, C. (2012). "Linguistic-based emotion analysis and recognition for measuring consumer satisfaction: an application of affective computing," *Inf. Technol. Manag.*, *13(4)*, 321–332.

[2] Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). "Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment," *Proc. Fourth Int. AAAI Conf. Weblogs Soc. Media*, pp. 178–185.

[3] Li, Y.-M., & Li, T.-Y. (2013). "Deriving market intelligence from microblogs," *Decis. Support Syst.*, *55(1)*, 206–217.

[4] Zhang, K., Downey, D., Chen, Z., Xie, Y., Cheng, Y., Agrawal, A., Liao, W.K. & Choudhary, A. (2013). "A probabilistic graphical model for brand reputation assessment in social networks". In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)* pp. 223-230.

[5] Nagamma, P., Pruthvi, H. R., Nisha, K. K., & Shwetha, N. H. (2015). "An improved sentiment analysis of online movie reviews based on clustering for box-office prediction," In *International Conference on Computing, Communication & Automation*, pp. 933–937.

[6] Nasraoui, O. (2008). "Book Review : Web Data Mining - Exploring Hyperlinks , Contents , and Usage Data," *SIGKDD Explor.*, *10(2)*, 23–25.

[7] Ravi, K., & Ravi, V. (2015). "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Syst.*, *89*, 14–46.

[8] Akter, S. & Wamba, S. F. (2016). "Big data analytics in E-commerce: a systematic review and agenda for future research," *Electron. Mark.*, *26(2)*, 173–194.

[9] Alaei, A. R., Becken, S., & Stantic, B. (2017). "Sentiment Analysis in Tourism: Capitalizing on Big Data," *J. Travel Res.*, *58(2)*, 175–191.

[10] Sehgal, D., & Agarwal, A. K. (2018). "Real-time Sentiment Analysis of Big Data Applications Using Twitter Data with Hadoop Framework. In *Soft computing: theories and applications*, pp. 765-772. Springer, Singapore, 2018.

[11] Wan, Y., & Gao, Q. (2016). "An Ensemble Sentiment Classification System of Twitter Data for Airline Services Analysis," *Proc. - 15th IEEE Int. Conf. Data Min. Work. ICDMW 2015*, pp. 1318–1325.

[12] Al-Smadi, M., Al-Ayyoub, M., Jararweh, Y., & Qawasmeh, O. (2019). "Enhancing Aspect-Based Sentiment Analysis of Arabic Hotels' reviews using morphological, syntactic and semantic features," *Inf. Process. Manag.*, *56(2)*, pp. 308–319.

[13] Chen, W., Zong, L., Huang, W., Ou, G., Wang, Y., & Yang, D. (2011). "An Empirical Study of Massively Parallel Bayesian Networks Learning for Sentiment Extraction from Unstructured Text," in *Proceedings of the 13th Asia-Pacific Web Conference on Web Technologies and Applications*, pp. 424–435.

[14] Airoldi, E., Bai, X. & Padman, R., (2004). Markov blankets and meta-heuristics search: Sentiment extraction from unstructured texts. In *International Workshop on Knowledge Discovery on the Web* (pp. 167-187). Springer, Berlin, Heidelberg.

[15] Pardalos, P.M., Du, D.Z. & Graham, R.L. eds., (2013). *Handbook of combinatorial optimization*. New York: Springer..

[16] Orimaye, S. O.(2013). "Sentiment augmented Bayesian network," *Conf. Res. Pract. Inf. Technol. Ser.*, *146*, pp. 89–98.

[17] Orimaye, S. O., Pang, Z. Y. & Setiawan, A. M. (2016). "Learning sentiment dependent Bayesian Network classifier for online product reviews," *Informatica*, *40*, 225, 2016.

[18] Ren, F., & Kang, X. (2013). "Employing hierarchical Bayesian networks in simple and complex emotion topic analysis," *Comput. Speech Lang.*, *27*, 943–968.

[19] Lane, P. C. R., Clarke, D., & Hender, P. (2012). "On developing robust models for favourability analysis: Model choice, feature sets and imbalanced data," *Decis. Support Syst.*, *53(4)*, 712–718.

[20] Kang, M., Ahn, J., & Lee, K. (2018). "Opinion mining using ensemble text hidden Markov models for text classification," *Expert Syst. Appl.*, *94*, 218–227.

[21] Grosan, C. & Abraham, A. (2011). *Intelligent systems*, *17*, 261-268. Berlin: Springer.

[22] Gutiérrez, L., Bekios-Calfa, J. & Keith, B. (2018). A review on bayesian networks for sentiment analysis. In *International Conference on Software Process Improvement* (pp. 111-120). Springer, Cham.

[23] John, G.H. & Langley, P. (2013). Estimating continuous distributions in Bayesian classifiers. *arXiv preprint arXiv:1302.4964*.

[24] Driver, E. & Morrell, D. (1995). "Implementation of Continuous Bayesian Networks Using Sums of Weighted Gaussians," *Proc. Elev. Conf. Uncertain. Artif. Intell.*, pp. 134–140.

[25] Friedman, N. and Goldszmidt, M., 1996, July. Discretizing continuous attributes while learning Bayesian networks. In *ICML* (pp. 157-165).

[26] Cooper, G. F. (1990). "The computational complexity of probabilistic inference using bayesian belief networks," *Artif. Intell.*, *42(2)*, 393–405.

[27] Heckerman, D. (2008). "A tutorial on learning with Bayesian networks," *Stud. Comput. Intell.*, *156*, 33–82.

[28] Cox, D.R. & Snell, E.J. (2018). *Analysis of binary data*. Routledge

[29] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R. & Lin, C.J. (2008). LIBLINEAR: A library for large linear classification. *The Journal of machine Learning research*, *9*, 1871-1874.

[30] Genkin, A., Lewis, D. D., & Madigan, D. (2007). "Large-scale bayesian logistic regression for text categorization," *Technometrics*, *49(3)*, 291–304.

[31] Guerin, A. (2016). "Using Demographic Variables and In-College Attributes to Predict Course-Level Retention for Community College Spanish Students," Northcentral University: Scottsdale, AZ, USA.

[32] Kaufmann, S., (2010). *CUBA: Artificial conviviality and user-behaviour analysis in web-feeds* (Doctoral dissertation, University of Luxembourg, Luxembourg, Luxembourg).

[33] Pearson, E. S. (1925). "Bayes' Theorem, Examined in the Light of Experimental Sampling," *Biometrika*, *17(3/4)*, 388, 1925.

[34] Qu, Z. Song, X., Zheng, S., Wang, X., Song, X., & Li, Z. (2018). "Improved Bayes Method Based on TF-IDF Feature and Grade Factor Feature for Chinese Information Classification.," *In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 15–17 January 2018;* pp. 677–680.

[35] Vapnik, V. & Chervonenkis, A. Y. (1964). "A class of algorithms for pattern recognition learning," *Avtomat. i Telemekh*, *25(6)*, 937–945.

[36] Magerman, D. M. (1995). “Statistical Decision-Tree Models for Parsing,” in *33rd Annual Meeting of the Association for Computational Linguistics*, pp. 276–283.

[37] Quinlan, J. R. (1986). “Induction of decision trees,” *Mach. Learn., 1(1),* 81–106.

[38] Kowsari, K., Meimandi, K. J. , Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). “Text classification algorithms: A survey,” *Inf., 10(4)*, 1–68.

[39] Ho, T.K. (1995) “Random decision forests”. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada 14–16 August 1995; Volume 1, pp. 278–282.

[40] Sutskever, I., Martens, J., & Hinton, G. (2011). “Generating text with recurrent neural networks,” *Proc. 28th Int. Conf. Mach. Learn. ICML 2011*, pp. 1017–1024.

[41] Hochreiter, S., & Schmidhuber, J. (1997). “Long Short-Term Memory,” *Neural Comput., 9(8)*, 1735–1780.

[42] Maas, A. L., Daly, R. E., Pham, P. T. , Huang, D., Ng, A. Y., & Potts, C. (2011). “Learning Word Vectors for Sentiment Analysis,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. pp. 142–150.

[43] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. , & Witten, I. H. (2009). “The {WEKA} data mining software: an update,” *SIGKDD Explor., 11(1),* 10–18.

[44] Buitinck, L. *et al.* (2013) ‘API design for machine learning software: experiences from the scikit-learn project’, pp. 1–15. Available at: http://arxiv.org/abs/1309.0238.

[45] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. & Ghemawat, S. (2016). “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. *arXiv preprint arXiv:1603.04467*.

[46] Cooper, G. F. & Herskovits, E. (1990). “A Bayesian method for constructing Bayesian belief networks from databases,” *Proc. Conf. Uncertain. AI, 91(5),* 86–94.

[47] Buntine, W. (1996). “A guide to the literature on learning probabilistic networks from data,” *IEEE Trans. Knowl. Data Eng., 8(2)*, 195–210.

[48] Bouckaert, R. R. (1995). Bayesian Belief Networks: From Construction to Inference. In *Dissertation*. Retrieved from https://dspace.library.uu.nl/handle/1874/845

[49] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2016). “Distributed Representations of Words and Phrases and their Compositionality Tomas,” *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 1389–1399.