A Study the Adaptive Big Data Model for Sentiment Analysis

Priyesh Upadhyay[1]*, Dr. Ravindra Tiwari[2]

[1] Research Scholar, LNCT University, Bhopal

[2] Associate Professor, LNCT University, Bhopal

Abstract- This study proposed an adaptable model for topic-level SA of social media data using deep learning. A topic modeling-based strategy, live latent semantic indexing with regularization constraint, was developed for topic extraction from streaming data. This study uses a long short-term memory network with a topic-level attention mechanism for SA. The proposed model's meat and potatoes are its ability to extract topics from streaming sentences from social media platforms in an online manner via adaptive updates to the model for new incoming streaming data, followed by topic-level SA on those streaming sentences to determine the polarity of the detected topics. The online latent semantic indexing topic-coherency score produced demonstrates the effectiveness of the suggested method for topic detection on real-time data. The SemEval-2017 dataset is used to train the model for SA.

Keywords- Sentiment Analysis, Social Media, LSTM Network Layers, Deep Learning

---------------------------------X---------------------------------

INTRODUCTION

Sentiment analysis (SA) is the practice of gleaning meaning from the words people use to express their thoughts, feelings, and attitudes on social media. There are several other ways in which this text data could be found, including reviews, blogs, news, & comments. Many businesses all around the world have adopted the ability to derive insights from this kind of data. Social media sites like Twitter, Facebook, Instagram, & Tumbler provide people the power to share stuff they are interested in, which can then be further evaluated utilizing analytical tools to offer insights & real-world solutions to problems we face every day. Due to the fact that it sheds light on sentiments, social media analysis is sometimes referred to as sentiment analysis or opinion mining. There are numerous methods and technologies that can be utilized for sentiment analysis. Prior to that, it's crucial to understand the topic on which you intend to use the sentiment analysis model.

DEEP LEARNING METHODS FOR SA

To perform sentence-level sentiment analysis, we have used following deep learning models – deep feed forward NN and LSTM network. Initially, pre-processing is performed, as shown in figure 1.

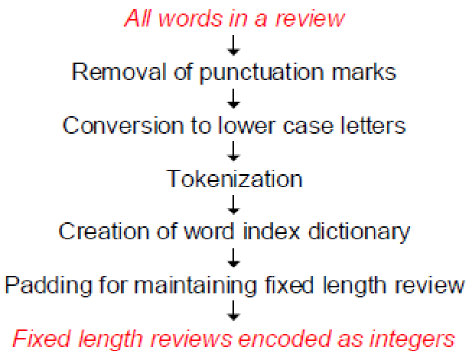

All words in a review
Removal of punctuation marks
Conversion to lower case letters
Tokenization
Creation of word index dictionary
Padding for maintaining fixed length review
Fixed length reviews encoded as integers

Figure 1: Pre-processing of input reviews

Initially, pre-processing of input reviews has been performed by removing the punctuation marks, converting them in lower case letters, and tokenizing the words. For represent a review as an ordered sequence of integers, word index dictionary is created, and for maintaining a uniform length of a review, padding is applied. We encoded positive labels with '1' and negative with '0'.

Sentiment Analysis using Deep Feedforward Neural Network

As shown in figure 2, deep feedforward NN consists of 1) embedding layer, 2) global average pooling layer, and 3) 6 dense layers. Embedding layer maps static reviews into embedded vectors. Global average pooling, as shown in figure 5.3, acts as a structural regularizer to avoid overfitting. Rectifier linear unit (ReLU) has been used with dense layers, excluding the last layer as an activation function.

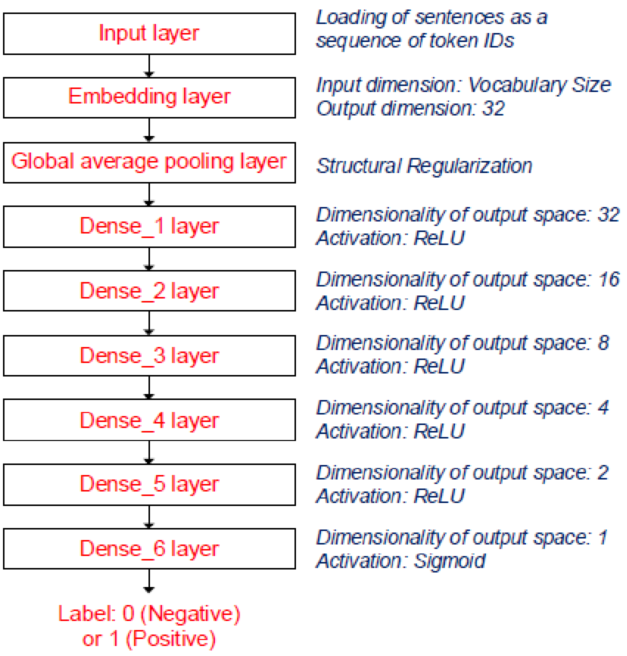

Input layer
Loading of sentences as a sequence of token IDs
Embedding layer
Input dimension: Vocabulary Size
Output dimension: 32
Global average pooling layer
Structural Regularization
Dense_1 layer
Dimensionality of output space: 32
Activation: ReLU
Dense_2 layer
Dimensionality of output space: 16
Activation: ReLU
Dense_3 layer
Dimensionality of output space: 8
Activation: ReLU
Dense_4 layer
Dimensionality of output space: 4
Activation: ReLU
Dense_5 layer
Dimensionality of output space: 2
Activation: ReLU
Dense_6 layer
Dimensionality of output space: 1
Activation: Sigmoid
Label: 0 (Negative) or 1 (Positive)

Figure 2: Deep feedforward NN model

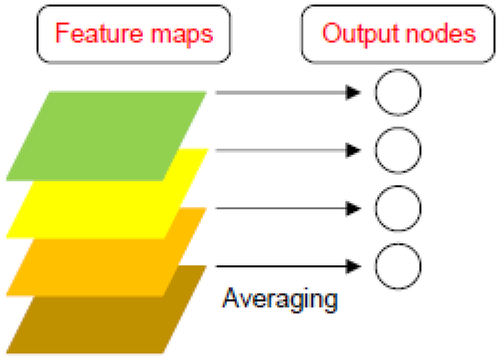


Figure 3: Global average pooling

SA utilizing LSTM Network

The configuration of the LSTM network layers is depicted in figure 4. It handles long-term dependencies and uses gate vectors to process the data and able to control information passing along the sequence. The workflow of LSTM modules for sentiment analysis is depicted in figure 5. The inputs to LSTM are given as , $ht{-}1$, $ct{-}1$.

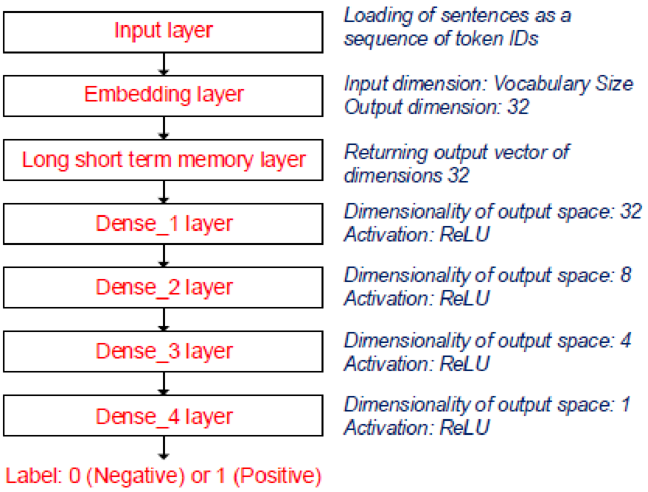

Input layer
Loading of sentences as a sequence of token IDs
Embedding layer
Input dimension: Vocabulary Size
Output dimension: 32
Long short term memory layer
Returning output vector of dimensions 32
Dense_1 layer
Dimensionality of output space: 32
Activation: ReLU
Dense_2 layer
Dimensionality of output space: 8
Activation: ReLU
Dense_3 layer
Dimensionality of output space: 4
Activation: ReLU
Dense_4 layer
Dimensionality of output space: 1
Activation: ReLU
Label: 0 (Negative) or 1 (Positive)

Figure 4: Configuration of LSTM network used for sentiment analysis

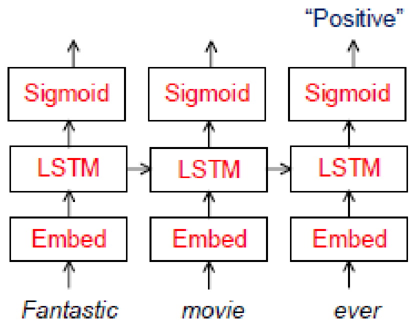


Figure 5: Sentiment analysis using LSTM model

Experimentation details of applying Deep Feedforward NN and LSTM network

IMDB Movie Review dataset has been used to analyze the sentiments. This dataset contains 50,000 reviews out of which 25,000 reviews have been used for training and 25,000 for testing. The models have been implemented in Python and TensorFlow environment with Keras API and executed on Google Compute Engine.

Results of applying Deep Feedforward NN and LSTM network

We checked the effectiveness of a deep feedforward NN for the task of in-domain SA by training & testing it on the IMDB dataset. Figures 6 and 7 depict the 88% validation accuracy and 30% loss achieved for the model, respectively. We checked the efficacy of the LSTM model for the task of out-of-domain SA by training on the IMDB dataset and testing on the Restaurant reviews dataset. We have implemented these deep architectures to evaluate the sample results on the standard dataset and to understand the working phenomena of LSTM over other Deep learning models so that we can configure it with our proposed adaptive model. Figures 6 and 7 depict the accuracy of 78% and loss of 29% respectively achieved for the LSTM model.

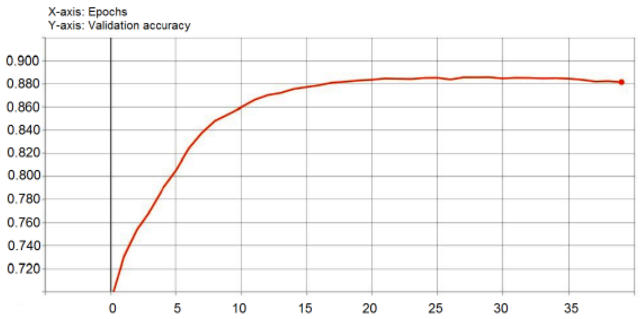


Figure 6: Validation accuracy obtained per epoch for deep feedforward NN

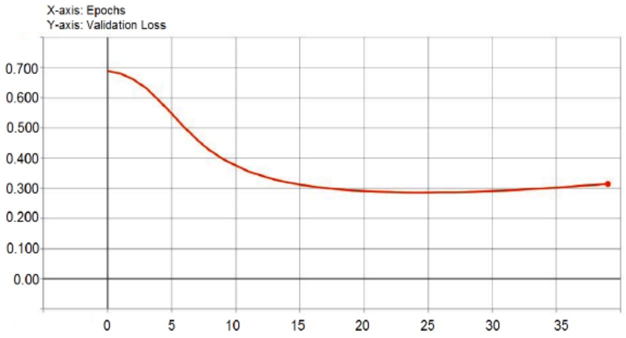


Figure 7: Validation loss obtained per epoch for deep feedforward NN

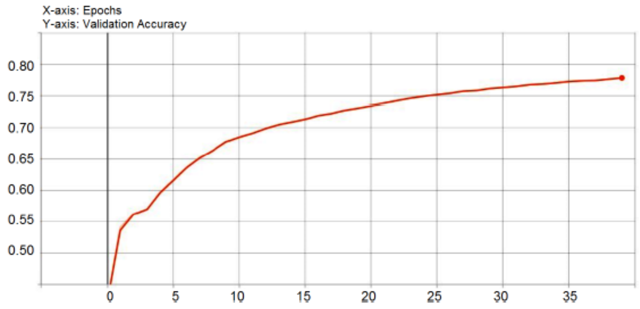


Figure 8: Validation accuracy obtained per epoch for LSTM network

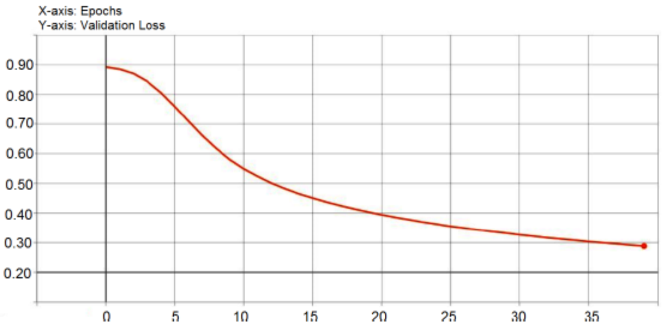


Figure 9: Validation loss obtained per epoch for LSTM network

Table 1: Comparison of results of applying Deep Feedforward NN and LSTM network

Parameters	Deep Feedforward NN (In-domain sentiment analysis)	LSTM Network (Out-of-domain sentiment analysis)
Validation Accuracy	88%	78%
Validation Loss	30%	29%
Training Dataset	IMDB	IMDB
Test Dataset	IMDB	Restaurants reviews

Table 1 compares the performance of deep feedforward neural network and LSTM network for the task of in-domain and out-of-domain SA, respectively.

METHODOLOGY OF ADAPTIVE BIG DATA MODEL FOR SA

In this part, the approach of the Adaptive Big Data Model for SA has been proposed. The proposed method is summarized in Figure 11. As per our assumption, each incoming sentence from a social networking platform is handled as a document. Figure 10 illustrates the workflow that was used to perform pre-processing and generate feature vectors. The approach is divided into 3 steps as the construction of feature vectors, topic detection utilizing online latent semantic indexing, & topic-level SA utilizing the LSTM network with topic-level attention. Essentially, the tokenization is conducted on each sentence from the streaming input to produce a list of words utilizing a spaCy library.

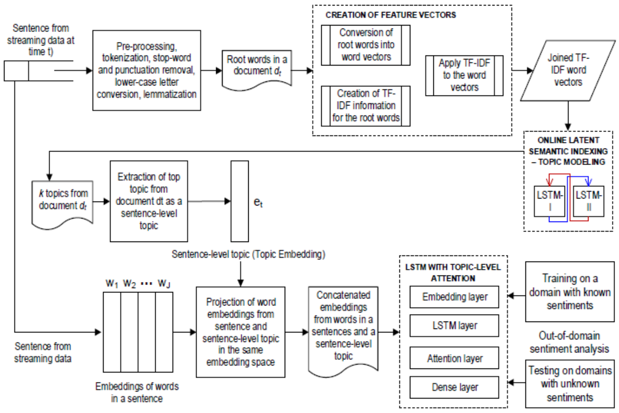

Sentence from streaming data at time t)
Pre-processing, tokenization, stop-word and punctuation removal, lower-case letter conversion, lemmatization
Root words in a document d_t
CREATION OF FEATURE VECTORS
Conversion of root words into word vectors
Creation of TF-IDF information for the root words
Apply TF-IDF to the word vectors
Joined TF-IDF word vectors
ONLINE LATENT SEMANTIC INDEXING – TOPIC MODELING
LSTM-I
LSTM-II
k topics from document d_t
Extraction of top topic from document dt as a sentence-level topic
e_t
Sentence-level topic (Topic Embedding)
W_1 W_2 ··· W_J
Sentence from streaming data
Embeddings of words in a sentence
Projection of word embeddings from sentence and sentence-level topic in the same embedding space
Concatenated embeddings from words in a sentences and a sentence-level topic
LSTM WITH TOPIC-LEVEL ATTENTION
Embedding layer
LSTM layer
Attention layer
Dense layer
Training on a domain with known sentiments
Out-of-domain sentiment analysis
Testing on domains with unknown sentiments

Figure 10: Proposed methodology of the adaptive big data model for sentiment analysis

The Gensim library is then utilized to clean up the text by removing punctuation, capitalization, & stop words. The etymological roots are obtained using lemmatization. By incorporating Word2vec word vectors with TF-IDF topic recognition characteristics, we hope to increase precision. We employed pre-trained vectors on a Google News dataset containing over 100 billion words. The vector has 300 dimensions. There are two primary justifications for integrating these separate capabilities. For one, topic modeling with TF-IDF captures the semantic relationship between words & returns the most relevant topics using a bag-of-words (BoW) model. The word's syntactic & semantic relationships are preserved thanks to the word2vec model. Consequently, we first generate 300-dimensional word vectors for each of the cleaned words using the Word2vec model from the Gensim package. After that, TF-IDF data is generated for each document using the Scikit-learn module.

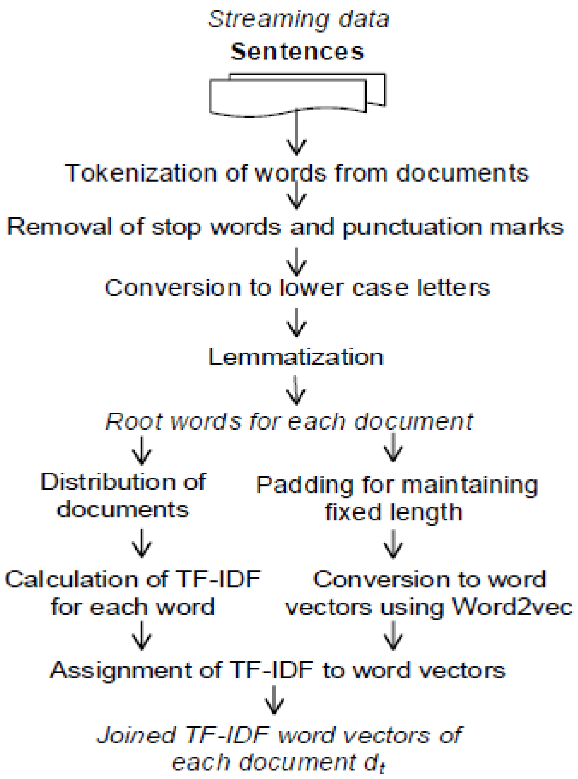

Streaming data
Sentences
Tokenization of words from documents
Removal of stop words and punctuation marks
Conversion to lower case letters
Lemmatization
Root words for each document
Distribution of documents
Padding for maintaining fixed length
Calculation of TF-IDF for each word
Conversion to word vectors using Word2vec
Assignment of TF-IDF to word vectors
Joined TF-IDF word vectors of each document d_t

Figure 11: feature vectors formation & Pre-processing

TOPIC-LEVEL ATTENTION BASED LSTM NETWORK FOR SA

If we define the embedding lookup produced by the Word2vec model as WRd|V|, where d is the dimension of the word embedding & |V| is the vocabulary size, then we can say that the Word2vec model produces embedding with the following properties. We can depict the input sentence's word embedding using the notation w1, w2,..., wJ, where wJ Rd. The embedded representations et of the topic recognized using online LSI for the given sentence, & topic embedding dimension det, are both shown below.

Within the embedding space, vectors representing word embedding & topic embedding could be multiplied, summed, & concatenated to form larger vectors. We concatenated the topic embedding with the word embedding's of a phrase to capture the effect of the topic on sentiment, as demonstrated by experiments in [S. Ruder 2016]. These concatenated vectors mapped in the same embedding space are then provided as input to the attention-based LSTM network.

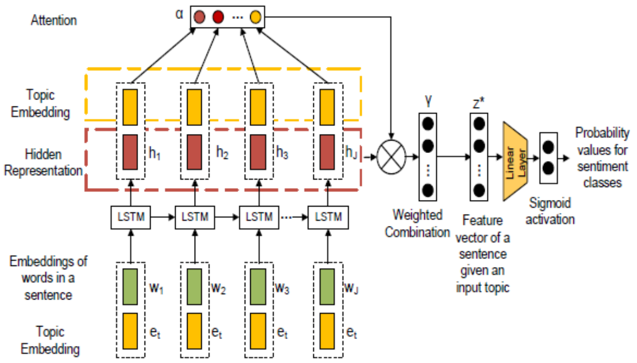


Figure 12: Topic-level attention LSTM network for sentiment analysis

EXPERIMENTATION DETAILS

The desired paradigm is implemented in Python. To get there, we used Keras API with TensorFlow as the backend & number of supplementary libraries, including spaCy, Scikit-learn, & Gensim. All of the tests have been conducted in Google's cloud environment, which consists of two virtual CPUs, seven & half gigabytes of RAM, & 64 terabyte persistent disk.

Datasets utilized

We created our datasets for topic detection in streaming phrases using the hashtags #facebook, #bitcoin, & #ethereum. Mendeley's research includes a publication of the datasets. In order to fine-tune the model for topic detection on the proposed datasets, we used random walk training. For the first round of adjustments, we utilized a week's worth of tweets from each of the three datasets. Statistics from the dataset used for tuning are displayed in Table 2.

Table 2: Dataset for initial fine-tuning for topic detection

Name of dataset	Approximate No. of Tweets used for initial fine-tuning (Tweets belonging to 1 week)
Facebook	48861
Ethereum	45861
Bitcoin	46494

Table 3: Dataset for topic-level sentiment analysis

Phase	Dataset	Positive	Negative	Total
Training	SemEval-2017 Task 4 Subtask B	14897	3997	18894
Testing	SemEval-2017 Task 4 Subtask B	2463	3722	6185
	Ethereum	1067	933	2000
	Bitcoin	1129	879	2000
	Facebook	756	1244	2000

We used the SemEval-2017 Task 4 Subtask B dataset [S. Rosenthal 2017] to train and evaluate our topic-level attention LSTM model for in-domain topic-level SA. We developed the model on the SemEval-

2017 Task 4 Subtask B dataset and evaluated it on three different datasets—Facebook, Ethereum, & Bitcoin—to determine how well it performed on tasks requiring SA outside of its original domain. To evaluate performance, we used the following procedures across all three datasets. During testing, we used a dataset consisting of continuously streaming phrases to feed our topic detection module, which then extracted the topic from the data. The suggested topic-level SAmodel is then used to calculate the sentiment score based on the extracted topic & phrase.

Training & Regularization

The LSTM network utilized for topic recognition & attention-based LSTM network utilized for topic-level SA both have their training & regularization procedures outlined below.

For LSTM network utilized for topic detection

Each embedded word vector has 300 dimensions in size, which is the default size for Word2vec output. For both of the LSTM networks shown in Fig. 12, we used truncated backpropagation through time (TBPTT) for topic detection because of the high cost of a single parameter update in BPTT. We employed the standard configuration of TBPTT (n1, n2), where n1=n2=100s, in accordance with the practices described in Williams 1990, in which the same number of time steps is performed to both the forward & backward passes. To boost performance, we introduce noise into the LSTM's initial state. The LSTM's forget gate's biases have been set to 1 for improved long-term memory. The latent dimension was maintained at 1024 elements long.

For attention-based LSTM network utilized for topic-level SA

Figure 13 depicts an end-to-end trained attention-based LSTM network learned utilising backpropagation. Size of embedded vectors is set to 300, latent dimension size is set to 1024, epochs are set to 30, and batch size is set to 64 in the LSTM model. In the last layer, we compared the entropy of the predicted labels to the entropy of the actual labels to determine how strongly each was associated with the sentiment. The Adam optimizer was employed to achieve this peak performance. The l2 norm penalty was set to 0.01, making it the default for the activity regularizer.

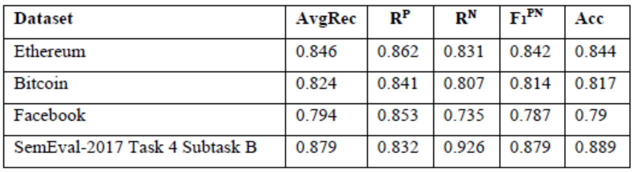
RESULTS

Both quantitative & qualitative interpretations of the findings are explored. Recall, average recall, the macro-average F1 score, & accuracy are used as performance indicators to assess the suggested model quantitatively. For qualitative analysis, the performance is evaluated in terms of scalability to support streaming data from social media networks.

Quantitative analysis

The suggested model's efficacy has been tested for both within-domain and cross-domain SA. We used the SemEval-2017 Task 4 Subtask B dataset for both training and testing the model for in-domain SA. The model has been evaluated on three datasets: Ethereum, Bitcoin, & Facebook obtained through the Twitter API using the #ethereum, #bitcoin, & #facebook hashtags, respectively, to assess its ability to perform out-of-domain SA. Metrics including the macro-average F1 score generated over positive & negative classes (F1PN), or accuracy (Acc) have been utilized to evaluate the model's performance on these four datasets.

Table 4: topic-level SA outcome of utilizing the proposed model to test datasets



Dataset	AvgRec	R^P	R^N	F_1^{PN}	Acc
Ethereum	0.846	0.862	0.831	0.842	0.844
Bitcoin	0.824	0.841	0.807	0.814	0.817
Facebook	0.794	0.853	0.735	0.787	0.79
SemEval-2017 Task 4 Subtask B	0.879	0.832	0.926	0.879	0.889

The outcomes of the topic-level SA are displayed in Table 4. Average recall was employed as the major metric in our tests, as described in the study [S. Rosenthal 2017]. The overall F1 score & degree of precision are only secondary indicators. It has been shown that the average recall measure is resistant to class imbalance [F. Sebastiani, 2015]. Since there is a greater emphasis on the average recall measure than on standard accuracy due to the uneven number of positive & negative classes in the SemEval dataset used for training (table 5.3), this dataset is highly recommended.

Qualitative Analysis

In terms of scalability, the effectiveness of the proposed model has been illustrated qualitatively, & metrics regarded to evaluate the model's scalability are the throughput in terms of topics detected per second, the average response time in seconds to handle the SA queries, & average time in milliseconds required to create feature vectors. The model was given data from the SemEval-2017 Task 4 Subtask B dataset in a continuous stream. Before anything else, we check how long it takes for the model to generate an average feature vector from input text (i.e., how long it takes to process a tweet). The typical amount of time in milliseconds required to generate a feature vector. We also monitor the model's efficiency by counting the number of topics it identifies per second relative to the rate at which tweets are being sent. Once a sentence and its topic have been submitted to the model, we then determine how long it will take to complete a SA query. To determine how long it takes for a server to respond on average to a sentiment analysis query, we change the rate at which these queries are received. Figure 13 demonstrates that up to 500 tweets per second, the average time in milliseconds required to create feature vectors remains constant. At this rate, tweets will be processed in real time. It then takes time to process the tweet's text. Figure 14 shows that throughput grows up to the point where the incoming tweet rate hits 700 per second. After this rate, the throughput plateaus, indicating that the model has hit its processing limit. It substantiates the processed model's ability to identify topics in streaming data at a pace of 700 tweets/second.

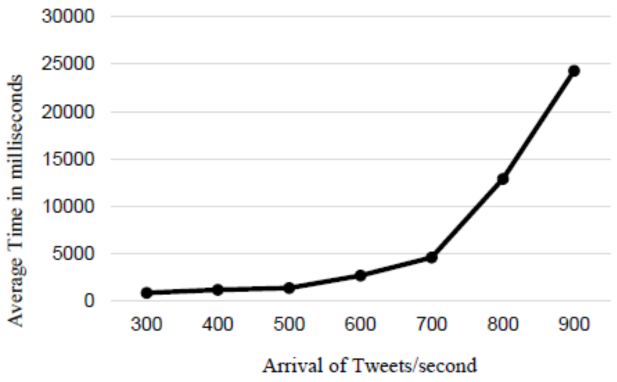


Figure 13: Average time in milliseconds for creation of feature vectors

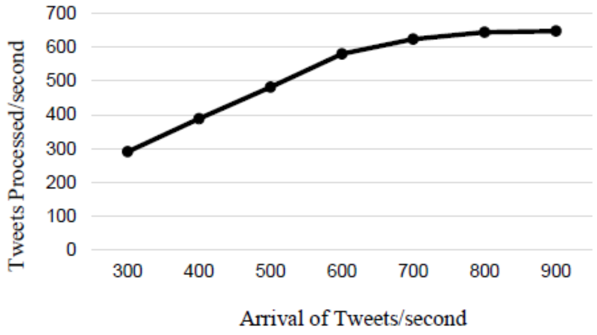


Figure 14: Throughput in seconds

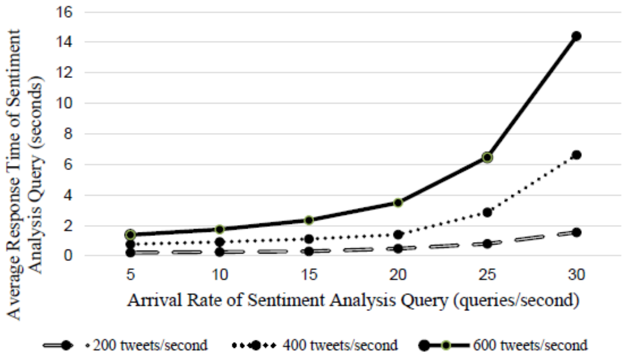


Figure 15: Average response time of SA query

Figure 15 illustrates how the influx of SA queries causes a shift in response times. We tested three different tweet-input rates (200, 400, & 600 tweets per second) & range of SAs query rates (5-30 inquiries per second) to determine the optimal pace. With a load of 30 inquiries per second & tweet arrival rate of 600 per second, the typical response time for SA queries is around 15 seconds.

These findings show that the proposed model is effective for topic-level SA & meets the requirements to facilitate online response at scale using streaming data.

CONTRAST WITH STATE-OF-THE-ART METHODS

By using Gaussian Nave Bayes as a baseline model for topic-level SA, we were able to get an average accuracy, F1PN score, & accuracy of 0.511, 0.528, & 0.542. To gauge how well the suggested model performs, it was pitted against the best models entered into subtask B of task 4 of the SemEval-2017 competition.

Table 5 summarizes the findings of a comparison between the proposed method and other, more advanced methods of topic-level SA. Figure 16 depicts the compared outcomes.

Table 5: Contrast with state-of-the-art topic-level SA approaches

Model	AvgRec	F_1^{PN}	Acc
BB_twtr [132]	0.882	0.890	0.897
DataStories [133]	0.856	0.861	0.869
Tweester [134]	0.854	0.856	0.863
funSentiment [135]	0.834	0.824	0.827
CrystalNest [136]	0.827	0.822	0.827
Gaussian Naïve Bayes (Baseline)	0.511	0.528	0.542
Topic-level attention LSTM network (Proposed)	0.879	0.879	0.889

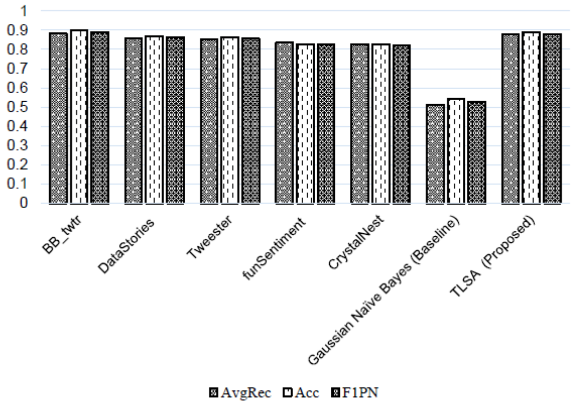


Figure 16: Contrast with state-of-the-art topic-level SA approaches

CONCLUSION

SA is an important part of NLP because it allows us to learn how people feel about many topics, including politics, products, & current events on a global scale. Combining topic modeling & SA yields insights into the most frequently discussed topics across different social media channels. Understanding how users feel about a detected issue can guide strategic decision-making and the development of new features. Relying on topic modeling & deep learning, we offer a method for topic-level SA. The suggested model is extensible due to the online nature of topic detection. The proposed method works well with the brief texts typically found on social media. On the SemEval 2017 dataset for in-domain SA, the model scored an average of 0.879 accuracy, while on the Ethereum, Bitcoin, & Facebook datasets, it scored 0.846, 0.824, & 0.794 accuracy, respectively, for the task of out-of-domain SA at a topic level. Based on the results achieved for scalability, the proposed model may serve as a suitable model to work in real-time sentiment analysis systems to process streaming data in an online way.

REFERENCES

1. C. Sun, L. Huang, and X. Qiu, “Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence,” in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 380–385.

2. Cambria E, Schuller B, Xia Y and Havasi C. New Avenues in Opinion Mining and Sentiment Analysis. IEEE Intelligent Systems 2013; 28(2): 15-21.

3. Cambria E, Song Y, Wang H and Howard N. Semantic Multi-Dimensional Scaling for Open-Domain Sentiment Analysis. IEEE Intelligent Systems 2012; 29(2): 44-51.

4. Castillo, Carlos, Mendoza, M. and Poblete, B. (2011). Information credibility on twitter. AMC, Proceedings of the 20th international conference on World Wide Web, pp. 40-58.

5. Chandrasekaran, G., Nguyen, T. N., & Hemanth D, J. (2021). Multimodal sentimental analysis for social media applications: A comprehensive review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11(5), e1415.

6. Chen PJ, Ding JJ, Hsu HW, Wang CY, Wang JC (2017) Improved convolutional neural network based scene classification using long short-term memory and label relations. In: 2017 IEEE international conference on multimedia & expo workshops (ICMEW), pp 429–434

7. Chen, L. C., Lee, C. M., & Chen, M. Y. (2020). Exploration of social media for sentiment analysis using deep learning. Soft Computing, 24(11), 8187-8197.

8. Chen, X., & Xie, H. (2020). A structural topic modeling-based bibliometric study of sentiment analysis literature. Cognitive Computation, 12(6), 1097-1129.

9. Cho K, Van Merrie¨nboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078

10. Chong, Dazhi, and Hui Shi. "Big data analytics: a literature review." Journal of Management Analytics 2, no. 3 (2015): PP 175-201.

11. Christian, H., Suhartono, D., Chowanda, A., & Zamli, K. Z. (2021). Text based personality prediction from multiple social media data sources using pre-trained language model and model averaging. Journal of Big Data, 8(1), 1-20.

12. Culotta, A. (2010). Towards detecting influenza epidemics by analyzing Twitter messages. In: Proceedings of the First Workshop on Social Media Analytics, ACM, pp. 115–122.

13. Cyril, C. P. D., Beulah, J. R., Subramani, N., Mohan, P., Harshavardhan, A., & Sivabalaselvamani, D. (2021). An automated learning model for sentiment analysis and data classification of Twitter data using balanced CA-SVM. Concurrent Engineering, 29(4), 386-395.

14. D. M. Blei and J. D. Lafferty, “Dynamic topic models,” in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 113–120

15. D. Ma, S. Li, X. Zhang, and H. Wang, “Interactive attention networks for aspect-level sentiment classification,” arXiv Prepr. arXiv1709.00893, 2017.

16. D. Shamanta, S. M. Naim, P. Saraf, N. Ramakrishnan, and M. S. Hossain, “Concurrent inference of topic models and distributed vector representations,” in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2015, pp. 441–457

17. D. Tang, B. Qin, and T. Liu, “Document modeling with gated recurrent neural network for sentiment classification,” in Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 1422–1432.

18. D. Tang, B. Qin, and T. Liu, “Learning semantic representations of users and products for document level sentiment classification,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural

Language Processing (Volume 1: Long Papers), 2015, pp. 1014–1023.

Corresponding Author

Priyesh Upadhyay*

Research Scholar, LNCT University, Bhopal