© DEC 2024 | IRE Journals | Volume 8 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I6-1712456

Studies on Development of Model for Sentiment Analysis with Ml: A Review

NAGAVELLI YOGENDER NATH[1], GATTU RAMYA[2], R. PRASANTH REDDY[3], K. MANI RAJU[4]
[1]*Assistant Professor, Department of Computer Science & Engineering (AI&ML), Sumathi Reddy Institute of Technology for Women, Hyderabad.*
[2]*Assistant Professor, Department of Computer Science & Engineering, Vignan Institute of Management and Technology for Women, Hyderabad.*
[3, 4]*Assistant Professor, Department of Computer Science & Engineering, Malla Reddy (MR) deemed to be University, Hyderabad.*

Abstract- Sentiment analysis refers to the automated process of identifying the underlying emotional tone conveyed in a text. This task has become increasingly crucial today due to the exponential growth of opinion-oriented text generation on various online platforms including social media and e-commerce websites. The process of sentiment analysis has evolved through different phases like the use of handcrafted rules, sentiment-oriented dictionaries, machine learning, deep learning, transformer models, and large language models. These different phases have enriched the field of sentiment analysis and enabled it to get better insights into text for sentiment analysis. One of the most impactful developments in sentiment analysis was the creation of embedding's for words or sentences, which are useful for representing unstructured data as structured data. These embedding are representing words or sentences as numerical vectors in a high-dimensional space and have proven effective in capturing the semantic relationships between words. By leveraging the power of embedding's, sentiment analysis models can understand the complexities of human language and provide more precise insights into people's emotions and opinions. However, word embedding's not inherently generated for sentiment analysis, which means that they are not inherently sentiment-oriented. Therefore, many researchers have focused on developing sentiment-oriented word embedding's for sentiment words, but they have overlooked the importance of intensity words such as "little", "high", and "extremely". These intensity words determine the level of sentiment expressed in text, which is particularly useful for fine-level sentiment analysis. To address this issue, a review proposes an intensity-aware word embedding development model and sentiment analysis with ML.

Index Terms—Sentiment analysis, Embedding, Machine learning methods, Hybrid methods

I. INTRODUCTION

The current version of the World Wide Web is referred to as "Web 2.0". It is distinguished by increased user ability to generate and share content [1]. The rise of platforms such as blogs, forums, and social media has increased the amount of user generated content on the internet. It is often referred to as the "data explosion." This explosion of data has had a major impact on natural language processing (NLP) because the user-generated content provides valuable insights into customer opinions, preferences, and satisfaction levels. It is useful for improving products and services, tracking brand reputation, and identifying potential issues. So there is an extensive need to explore these opinions. [2]

Sentiment analysis (SA), frequently called opinion mining, is the process of determining an individual's attitude, opinions, and emotions about a specific topic or the overall contextual polarity of text. It has its roots in NLP, which instructs computers how to process and understand the language of humans. During the very initial phases of NLP, researchers focused primarily on developing systems that could recognize the syntax and semantics of the language. However, with the explosion of user-generated content, there was a growing need to analyze the opinions and attitudes stated in the text.

SA field employs a variety of techniques to recognize and classify the sentiment conveyed in text data. Some of the commonly used techniques are described briefly as follows.

© DEC 2024 | IRE Journals | Volume 8 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I6-1712456

Machine learning methods: Machine learning (ML) [3] methods use algorithms to learn patterns in labeled data and apply these patterns to classify new, unlabeled data. In SA, ML methods commonly include training a model on a labeled collection of text data, where the sentiment has been manually annotated. This approach can subsequently classify unseen textual data as positive, negative, or neutral. ML methods are powerful and flexible, but they necessitate an enormous quantity of labeled data and can be challenging to interpret. Furthermore, the models' accuracy is heavily dependent on manually created feature vectors.

Hybrid methods: Hybrid methods [4] combine multiple techniques or models to achieve higher accuracy in SA. For example, a hybrid methodology could employ a rule-based mechanism for detecting words that carry sentiments and then use a lexicon-based system to assign a sentiment score to each word. This sentiment score may then be used as a feature in the ML model to classify the sentiment of the overall text. Hybrid methods leverage the strengths of multiple techniques but can be complex to implement and interpret.

Deep learning methods: Deep learning methods [5] are a type of ML that employ neural networks (NN) to model complex data relationships. In SA, deep learning methods may involve using Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) to classify text data. These models are highly effective for certain types of text data, particularly when there is a lot of contextual information that affects sentiment, but they demand an immense amount of labeled data and are computationally expensive.

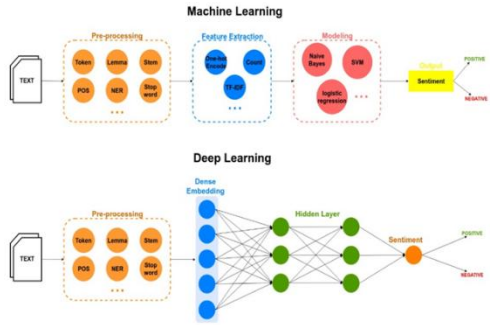

Figure 1: Sentiment analysis of Machine learning and Deep learning

Textual data is typically unstructured and contains a large amount of noise, which makes it challenging to analyze directly. Thus, text representation is required to turn unstructured textual input into a structured format that can be used for SA. This text representation is a critical component of NLP in ML and deep learning-based applications that deal with textual data. Text representation techniques are helpful for recognizing important text features, such as keywords and phrases, and representing them in a way that ML algorithms can easily analyze.

Early text representation techniques were based on bag-of-words models, which represented documents as a group of discrete words, ignoring the sequence in which they appeared. Bag-of-words models were simple and computationally efficient but suffered from several limitations. For example, they did not capture the semantic relationships between words and were unable to handle words which are not present in the vocabulary.

To address these limitations, researchers developed more sophisticated text representation techniques, such as n-grams and latent semantic analysis (LSA) [6]. N-grams are sequences of words of length n that are used to record the local context of words in a document. Latent Semantic Analysis (LSA), on the other hand, is a matrix factorization technique that uses single-value decomposition (SVD) to identify the underlying latent semantic structure of a document. Despite their improved performance over bag-of-words models, and n-grams; LSA suffered from several limitations, including the inability to handle large vocabularies and the difficulty of keeping track

© DEC 2024 | IRE Journals | Volume 8 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I6-1712456

of far-reaching dependencies between words in a document.

With the advent of deep learning, researchers developed new text representation techniques that were able to determine difficult relationships between words. These neural network-based techniques can learn hierarchical representations of text data. Word embedding's are a few of the most commonly used deep learning-based text representation techniques, which represent words as vectors in a space with high dimensions. A NN is trained on an extensive corpus to find word embedding's, with each word implied as a vector in identical space. Word embedding does can capture semantic connections among words, such as antonyms and synonyms; and can handle words that are not in the dictionary. The transformer model is another popular deep learning-based text representation technique [7]. The transformer model is based on a self-attention system that enables the model to flexibly concentrate on various segments of the input sequence. The transformer model can capture long-term dependencies between words in a document and has achieved cutting-edge performance on several NLP tasks, including SA and text categorization.

SA is related to the identification of the sentiment or opinion of a text. For sentiment identification, it employs a range of techniques, from rule-based methods to deep learning models. Every single one of these techniques has benefits as well as drawbacks, and the technique is selected by the particular demands of the SA task. In addition to selecting the model, text representation is another critical component of SA. Traditional text representation techniques like bag-of-words, n-grams, and LSA have been augmented by more sophisticated techniques based on deep learning, such as transformer models. Again, each method of text representation has pros and cons. The selection of a method is generally decided by considering the task at hand. For sentiment analysis, sentiment-oriented word embedding's are necessary. Therefore, a method that offers sentiment-oriented word embedding does can be selected.

II. REVIEW OF LITERATURE

The exploration begins with the various techniques employed for representing text in a numerical vector or matrix format. This transformation from unstructured text to structured data is pivotal in enabling computational analysis and interpretation of textual information. It includes diverse approaches utilized in converting textual data into a format responsive to computational analysis. From elementary static methods like vector space models, these approaches advance towards dynamic techniques like using deep learning models, enriching the process of text representation. Continuing exploration, this chapter investigates the techniques and models utilized in SA. SA, a key aspect of NLP, includes discerning and categorizing the sentiment conveyed in textual data. These approaches range from rule-based methods to machine learning and deep learning models. These approaches are used to classify the text at the binary level or fine-level sentiment analysis. Overall, text representation techniques and SA approaches are foundational tasks for SA, enabling researchers to extract significant insights from textual data and make informed conclusions in sentiment identification.

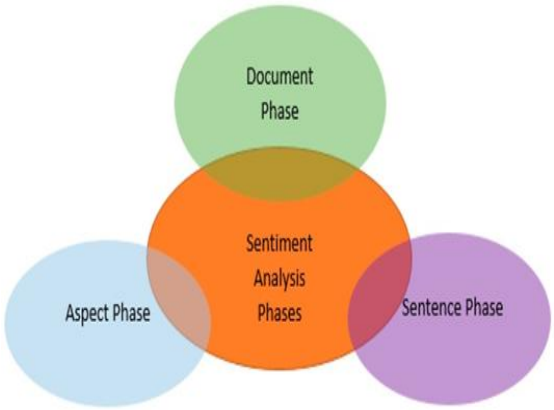

Figure 2: Sentiment analysis Phase

The need for text representation arises from the fact that natural language text data is discrete, unstructured, and high-dimensional. It complicates the processing for both classical and modern prediction-based algorithms which are designed to work with structured data. Text representation is the method of turning raw text data into a format that predictive models can understand and use. It is an essential step in many NLP applications like sentiment analysis. It enables various models to comprehend and make predictions depending on the text's meaning [8-9].

In general, the original input text is represented at the sentence or document level by considering the words present inside it. The text representation in a numerical vector or matrix is passed to the model for classification. This is the most crucial job in NLP as the result of the downstream task is dependent upon this representation. Previously, the text was represented by the Vector Space Model, Weighting System, Bag of Words, Distributional Representation, and Neural Network Representation, among several other representations [10-11].

Neural Network-based Representation

A neural network is a type of computational model that draws inspiration from the structure and functions of the human brain, often applied in various machine learning applications [12]. It includes a network of interconnected "neurons" organized in layers. Each neuron in a layer applies a series of mathematical operations to the input to produce an output, which is then passed to the next layer called the hidden layer. The network's last hidden layer's output is used to make predictions or decisions. The output layer makes the final decision. Shallow neural networks and deep neural networks are the two types of models. A shallow neural network is made up of three layers: an input layer, a hidden layer, and an output layer. If the number of hidden layers is two or more, the network is referred to as a deep neural network. Both of these models can be used to represent text. In general, the output layer represents the text by considering the context, domain, and meaning of the text.

The FastText architecture [13], which consists of a single linear layer followed by a softmax activation function, used to classify text into different categories. This linear layer takes the sum of the word vectors as input and produces a high-dimensional output vector. This output vector is then processed by the softmax function to generate a probability distribution over the various classes.

The model undergoes training by reducing the negative log-likelihood of the true class labels given the input text. The proposed architecture is simple, efficient, and can be utilized over massive datasets.

Deep Neural Network

Several different deep learning architectures can be used to represent text embeddings, including:

1. CNN: CNN is especially useful for analyzing grid-like information, such as images or text. It can be used to learn text embeddings by extracting features from input text data using convolutional filters.
2. RNN: It is specially adapted for managing sequential data like text. They are frequently used to learn text embeddings by sequentially processing the input text data and using the hidden state to represent the text embedding.
3. LSTM: These networks are well-suited for tasks like language understanding and machine translation by utilizing gates to selectively retain or forget information.
4. Hybrid architectures: Combining different architectures to leverage their unique properties can also be used to represent text embeddings. For example, A CNN-LSTM network can be employed to acquire characteristics from input text data using a CNN and then process the features sequentially using an LSTM.[14-15]

These different architectures can be used in combination with pre-training and finetuning for specific tasks to enhance the embedding effectiveness. Such architectures are popular for representing text due to the following reasons:

1. Representational capacity: Deep neural networks can acquire complex representations for text data, as they can use multiple layers of non-linear transformations to extract features from the input data. This can lead to better quality text embeddings for certain tasks.
2. Automatic Feature Engineering: End-to-end training of a deep neural network implies the network can map the input text directly to the desired text embeddings without the need for feature engineering.
3. Transfer learning: The ability to use already trained models on the massive corpus, which boosts the model's ability to perform a new task, using the concept of transfer learning. This can also save computational costs.
4. Handling Complexity: Deep neural networks are better equipped to handle highly complex and abstract data, in this way, they can handle tasks such as language understanding, machine translation, and embedding development.

5. Handling sequential data: RNNs and their variants (LSTM, GRU) are deep neural networks that are especially ideal for dealing with sequential data, such as text, which can make them useful for certain NLP activities like embedding development.

Machine Learning

Machine learning [3] methods automatically identify the sentiment of text without using handcrafted natural language rules. The sentiment assessment problem can be modeled as a classification problem, in which a machine learning model takes features of input text and generates output in terms of class, such as positive, negative, or neutral.

This approach includes training and prediction phases. For training, models accept the labeled text as input. Each input is categorized either as positive, negative, or neutral. The input text is converted into a vector, termed features, through the feature extractor.

Some of the features are uni-gram, bi-gram, tri-gram, n-gram, and term frequency, length of text, word position, and sentiment words. Along with these features, Parts-of- Speech [17], syntactic [18] and semantic [19] relations have been also used as rich natural language features.

The classification rules are automatically defined by the model based on the input features and labels. For defining rules, the model forms the internal representation of training data. The loss function, such as mean squared error, hinge loss and log loss, guides the model to define rules with the least amount of error. After training, the model predicts the sentiment of unseen text by converting it to a vector using a feature extractor and model-determined internal representation rules.

There are two types of models: probabilistic models and non-probabilistic models. The probabilistic models are the Nave Bayes Classifier, Bayesian Classifier, and Maximum Entropy Classifier. Non-probabilistic classifiers are Decision Trees (DT), Support Vector Machines (SVM), and Neural Networks (NN). In the past, researchers mostly used Nave Bayes Classifier, Maximum Entropy Classifier, and Support Vector Machines to classify sentiment. In addition to these types, ensemble machine learning combines models and makes decisions by combining the results obtained from individual models [20, 21].

III. TYPES OF SENTIMENT ANALYSIS

Binary SA and fine-grained SA are two different types to identify the sentiment expressed in the text. Binary sentiment classification involves classifying a text as either positive or negative, with no middle ground. This approach is relatively simple, as it only requires two categories for classification. It is useful for applications where a binary positive or negative distinction is sufficient. This approach is often used for tasks such as the SA of product reviews, where the objective is to find out whether the overall sentiment is positive or negative. Binary sentiment classification is also useful for tasks such as the SA of social media posts, where the aim is to rapidly identify whether a post expresses a positive or negative sentiment [22].

In contrast, fine-grained SA is a method for classifying text into more than two categories, such as positive, negative, and neutral. Sometimes, these categories are more specific, such as happy, sad, and angry. In many cases, fine-grained sentiment classification considers a scale from 1 to 5, 1 being the most negative and 5 being the most positive. This approach provides a more detailed understanding of the sentiment within a text by taking into account different shades and degrees of sentiment. Fine-grained SA is useful for applications where a more detailed understanding of sentiment is required, such as identifying specific emotions expressed in text [23].

Binary SA is a simple and fast method for determining whether text expresses a positive or negative sentiment, while fine-grained SA provides a more detailed understanding of the sentiment within the text by identifying different shades and degrees of sentiment.

IV. APPLICATIONS OF SENTIMENT ANALYSIS

Various domains utilize sentiment analysis by employing the methodologies, approaches, and techniques mentioned above. Some of the key applications of SA from the different domains are listed below.

© DEC 2024 | IRE Journals | Volume 8 Issue 6 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV8I6-1712456

SA finds diverse applications across industries, providing valuable insights into customer opinions, preferences, and emotions. In marketing, businesses utilize sentiment analysis to gauge consumer sentiment towards their products and brands.

Through the analysis of social media content, online reviews, and customer feedback, companies acquire actionable insights into customer satisfaction, pinpoint areas for enhancement, and customize marketing strategies to better connect with their intended audience.

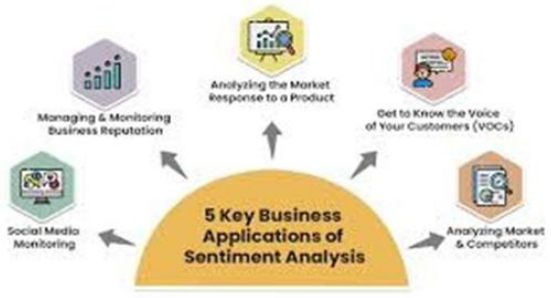

Figure 3: Application of Sentiment Analysis

In the realm of customer service, SA plays a crucial role in checking and handling customer experiences. By investigating customer interactions, such as emails, chat transcripts, and support tickets, companies can quickly identify and address issues, prioritize urgent inquiries, and ensure timely responses to customer inquiries and complaints. This proactive approach to customer service helps to enhance customer satisfaction, loyalty, and retention rates [24].

SA also finds applications in financial markets, where it is used to gauge investor sentiment and market trends. Through the examination of news articles, social media content, and financial reports, investors can extract valuable insights into market sentiment. They can discern potential market opportunities or risks and can render informed investment choices. SA can also be used for sentiment-based trading strategies, where investors leverage sentiment signals to predict market movements and optimize their investment portfolios.

Within the healthcare sector, SA is utilized to scrutinize patient feedback, reviews, and medical records, facilitating the extraction of insights into patient satisfaction levels, identifying areas for enhancement in healthcare services, and optimizing patient care outcomes. By understanding patient sentiment, healthcare providers can tailor their services to meet patient needs, improve patient experiences, and ultimately drive better health outcomes.

Furthermore, sentiment analysis finds applications in risk management, where it is used to assess sentiment toward companies, industries, or financial markets to identify potential risks and opportunities [25]. By exploring sentiment indicators, like news sentiment, investor sentiment, and market sentiment, risk managers can anticipate market movements, assess market sentiment, and make informed decisions to mitigate risks and optimize investment strategies.

Overall, sentiment analysis continues to find new and innovative applications across industries, highlighting its versatility and importance in extracting insights from textual data. It facilitates well-informed decision-making and improves corporate outcomes.

V. WORD EMBEDDING'S FOR FINE-LEVEL SENTIMENT ANALYSIS

In natural language, words such as "little", "more", and "extremely" can be used to indicate the intensity or degree of emotion. For instance, the word "little" might indicate a low intensity, while "more" might indicate a moderate intensity, and "extremely" might indicate a high intensity. These words are often called intensifiers and can be used to increase or decrease the intensity or magnitude of a sentence. For example, "I am a little tired" might indicate a low intensity of tiredness, while "I am extremely tired" would indicate a high intensity of tiredness. Similarly, "I am more tired" would indicate a moderate level of intensity of tiredness. Intensity words such as "more", "little", and "extremely" are important in sentiment analysis because they provide additional information about the strength or degree of the sentiment expressed in a text. Sentiment analysis typically assigns a positive, negative, or neutral score to a text, but the use of intensity words can add more granularity to this score. For example, a sentence that contains "more" might be assigned a more positive score than a sentence that

contains “little”, even if both sentences contain the same sentiment-bearing words. In addition, intensity words can also be used to detect sarcasm or irony, which can be hard to detect using classical SA techniques. For example, a sentence such as “I just love spending hours stuck in traffic” is likely to be sarcastic and convey a negative sentiment, but traditional sentiment analysis might assign a positive sentiment because of the word “love”. The use of intensity words can help to better identify the true sentiment in such cases. So, intensity words can also help to identify the emotional state of the speaker or writer, which can provide additional insights into the text [26].

Most of the literature on sentiment analysis has traditionally concentrated on sentiment-bearing words (also known as opinion words or seed words) rather than intensity words because sentiment-bearing words are more directly related to the sentiment of a text. Sentiment-bearing words, such as “good” and “bad”, directly indicate whether a text expresses a positive, negative, or neutral sentiment, while intensity words like “more” or “extremely” add information about the strength or degree of sentiment. Additionally, most of the early sentiment analysis research focused on binary classification (positive vs negative), which does not require the use of intensity words, but rather only the presence of positive or negative seed words.

However, in recent years, sentiment analysis has advanced to more complex tasks, such as multi-class classification, aspect-based sentiment analysis, and sentiment intensity detection, which require the use of intensity words. The research in this area is growing. The method used for developing intensity-aware word representations of intensity words. These representations are useful for identifying the sentiment of a given piece of text.

VI. CONCLUSION

This research has made a noteworthy contribution to the current literature by introducing and establishing a novel intensity-aware embedding development model that is useful for fine-grained SA. This model considers various levels of intensities of words while developing word embedding useful for sentiment analysis and has addressed the research gaps identified. The ultimate goal is to include the intensity of words while developing word embedding and providing a foundation for further fine-level analysis. This research aims to address challenges in the identification of sentiment from the vast amount of opinionated data automatically. To accomplish this, the current research work introduced several methods, including an intensity-aware embedding development model, global sentiment-oriented text generation with the local dynamic representation of text.

REFERENCES

[1] A. Yenkikar, N. Babu and S. Sangve, “R-SA: A Rule-based Expert System for Sentiment Analysis”, 2019 IEEE Pune Section International Conference Pune, India, 2019, pp. 1-7, doi: 10.1109/PuneCon46936.2019.9105682.

[2] S. Islam, X. Dong, “Domain-specific Sentiment Lexicons Induced from Labeled Document”, Proceedings of the 28th International Conference on Computational Linguistics, pages 6576–6587, Barcelona, Spain (Online), 2020. available at: https://aclanthology.org/2020.coling-main.578.pdf.

[3] C. Dang, “Hybrid Deep Learning Models for Sentiment Analysis”, Hindawi, Vo. 2021, https://doi.org/10.1155/2021/9986920

[4] L. Zhu, M. Xu, Y. Bao, Y. Xu, X. Kong, “Deep learning for aspect-based sentiment analysis: a review”, PeerJ Computer Science, 10.7717/peerjcs. 1044, 2022.

[5] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, M. Zhou, “Sentiment Embeddings with Applications to Sentiment Analysis”, IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 2, Feb 2016.

[6] L. Yu, K. Li, X. Zhang, “Refining Word Embeddings Using Intensity Scores for Sentiment Analysis”, in IEEE Transactions on Audio, Speech, and Language Processing, Vol. 26, No. 3, March 2018.

[7] R. Zhao and K. Mao, “Topic-Aware Deep Compositional Models for Sentence Classification”, IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 25, no. 2, Feb. 2017.

[8] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, “Improving language understanding by generative pre-training”, 2018.

[9] M.Joshi, D. Chen, Y. Liu, D.. Weld, L. Zettlemoyer, O. Levy, “Spanbert: improving pre-training by representing and predicting spans”, arXiv preprint,2020, arXiv:1907.10529

[10] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding”, arXiv preprint, 2020, https://arxiv.org/abs/1906.08237

[11] Nandipati Sai Akash, Naveen Sai Bommina, Uppu Lokesh, Hussain Syed, Syed Umar, "Optimized Block Chain-Enabled Security Mechanism for IoT Using Ant Colony Optimization", International Journal on Recent and Innovation Trends in Computing and Communication, (2023), 11(10), 1226–1233.

[12] D. Yan, B. Hu and J. Qin, “Sentiment Analysis for Microblog Related to Finance Based on Rules and Classification”, 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 119-126, 2018.

[13] Mohammed Kasri, Anas El-Ansari, Mohamed El Fissaoui, Badreddine Cherkaoui, Marouane Birjali, Abderrahim Beni-Hssane, “Public sentiment toward renewable energy in Morocco: opinion mining using a rule-based approach”, Social Network Analysis and Mining, Volume 13, Issue 1, 2023.

[14] Kousik Barik and Sanjay Misra, “Analysis of customer reviews with an improved VADER lexicon classifier”, Journal of Big Data, Volume 11, Issue 1, 2024.

[15] Naveen Sai Bommina, Nandipati Sai Akash, Uppu Lokesh, Dr. Hussain Syed, Dr. Syed Umar, "A Hybrid Optimization Framework for Enhancing IoT Security via AI-based Anomaly Detection", International Journal on Recent and Innovation Trends in Computing and Communication, (2023) ISSN: 2321-8169 Volume: 11 Issue: 3.

[16] Wei Lin and Li-Chuan Liao, “Lexicon-based prompt for financial dimensional sentiment analysis”, Expert Systems with Applications, Volume 244, 2024.

[17] B. Pang and L. Lee, “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts”, in Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics, pp. 115–124., 2004.

[18] Jacqueline Kazmaier, Jan H. van Vuuren, “The power of ensemble learning in sentiment analysis”, Expert Systems with Applications, Volume 187, 2022.

[19] Naveen Sai Bommina , Nandipati Sai Akash, Uppu Lokesh , Dr. Hussain Syed , Dr. Syed Umar, "Privacy-Preserving Federated Learning for IoT Devices with Secure Model Optimization", International Journal of Communication Networks and Information Security (IJCNIS), (2021), 13(2), 396–405.

[20] Ankit and Nabizath Saleena, “An Ensemble Classification System for Twitter Sentiment Analysis”, Procedia Computer Science, vol. 132, pg. 937- 946, 2018.

[21] P. Liu, X. Qiu, X. Huang, “Recurrent Neural Network for Text Classification with Multi-Task Learning”, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), pg.2873-2879, 2016.

[22] Uppu Lokesh, Naveen Sai Bommina, Nandipati Sai Akash, Dr. Hussain Syed, Dr. Syed Umar, "Designing Energy-Efficient and Secure IoT Architectures Using Evolutionary Optimization Algorithms", International Journal of Applied Engineering & Technology, Vol. 4 No.2, September, 2022.

[23] D. Tang, B. Qin, F. Wei, L. Dong, T. Liu, M. Zhou, “A Joint Segmentation and Classification Framework for Sentence Level Sentiment Classification”, IEEE/ACM Transaction on Audio, Speech, and Language Processing, Vol. 23, no. 11, Nov 2015.

[24] D. Yin, T. Meng, K. Chang, “SentiBERT: A Transferable Transformer- Based Architecture for Compositional Sentiment Semantics”, arXivpreprint, 2020, arXiv:2005.04114v4

[25] Naveen Sai Bommina , Nandipati Sai Akash, Uppu Lokesh , Dr. Hussain Syed , Dr. Syed Umar, "Multi-Objective Genetic Algorithms for Secure Routing and Data Privacy in IoT Networks", International Journal of

Communication Networks and Information Security (IJCNIS), (2020), 12(3), 632–643.

[26] Mr. Dikshendra Daulat Sarpate, and Dr. B.G Nagaraja, "CONVOLUTION NEURAL NETWORK-BASED SPEECH EMOTION RECOGNITION USING MFCCS", International Journal of Communication Networks and Information Security, 2023/12/10