IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, February 2024

Decoding Sentiments: Virtue or Vice through Multilingual Paragraph Analysis

Vishal U[1], Veena M V[2], Poornima R M[3]
Students, Department of Information Science and Engineering[1,2]
Assistant Professor, Department of Information Science and Engineering[3]
Global Academy of Technology, Bangalore, India

Abstract: *This work provides a comprehensive overview of recent developments in sentiment analysis methodologies. It explores innovative approaches, including the integration of rule-based sentiment dictionaries, machine learning techniques, and deep learning solutions for financial sentiment analysis. Emphasis is placed on key preprocessing steps such as tokenization, lowercasing, stop words removal, and punctuation elimination. Feature extraction techniques like Bag-of-Words, Word2Vec, and TF-IDF are discussed, highlighting their roles in representing textual information. The abstract delves into model selection, covering traditional machine learning models like Naive Bayes, Support Vector Machines, and Random Forests, as well as deep learning models such as Recurrent Neural Networks, Long Short-Term Memory networks, and BERT. The abstract explains these algorithms in detail, emphasizing their application in sentiment analysis. Training the model through supervised learning and evaluating its performance using metrics like accuracy, precision, recall, and F1 score are outlined. Additionally, a structured approach to paraphrasing is introduced, underlining its significance in creating meaningful representations of text.*

Keywords: sentiment analysis, preprocessing, deep learning, paraphrasing

I. INTRODUCTION

Natural language processing (NLP) is a method to systematically analyze and extract information from textual data, playing a vital role in artificial intelligence. Integrating various disciplines such as linguistics, computer science, and electrical engineering, NLP aims to enhance computational capabilities related to human language. Within the realm of NLP, sentiment analysis of text holds significance, involving the analysis and classification of sentiments and attitudes within subjective texts. Sentiment analysis (SA), a branch of natural language processing (NLP), utilizes machine learning techniques to extract relevant information from text data, discerning views, emotions, and attitudes. The primary goal is to categorize text polarity, determining whether it conveys positive, negative, or neutral sentiment. Machine learning methods, including supervised, unsupervised, and deep learning, play an important role in sentiment analysis, leveraging labelled datasets for training models or uncovering patterns and clusters in unlabelled data. Optimisation methods, such as gradient descent and genetic algorithms, enhance the precision of sentiment analysis models. Deep learning techniques, exemplified by the Deep-Recurrent Neural Network (D-RNN) introduced in this study, contribute to accurate sentiment analysis across various datasets. This application has gained prominence, especially with the surge in online expression through social media and other platforms. Researchers are focused on refining the accuracy and efficiency of text sentiment analysis to meet the demands of the era of big data.

II. RELATED WORK

In the realm of sentiment analysis, diverse studies offer innovative approaches to address challenges and enhance the accuracy of sentiment classification models.

Hao Liu, Xi Chen, and Xiaoxiao Liu[1] contribute to this field with a comprehensive study comparing rule-based sentiment dictionaries and machine learning methods. Their work proposes a novel approach that combines both techniques, achieving an impressive accuracy rate of 82.1%. Leveraging advanced techniques like BERT-large-cased and optimization algorithms such as stochastic gradient descent, their model exhibits effectiveness across various

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, February 2024

datasets, including Twitter, Restaurant, and Laptop. This study not only addresses the limitations of existing methods but also significantly improves sentiment analysis accuracy and efficiency.

Another noteworthy contribution comes from The paper "Deep-Sentiment: An Effective Deep Sentiment Analysis Using a Decision-Based Recurrent Neural Network (D-RNN)" by Putta Durga and Deepthi Godavarthi[2], from the School of Computer Science and Engineering at VIT-AP University, presents an integrated model for sentiment analysis. The approach combines techniques such as the pre-trained BERT-large-cased model, stochastic gradient descent optimization, and deep sentiment analysis based on a Decision-Based Recurrent Neural Network (D-RNN). The experiments conducted on Twitter, Restaurant, and Laptop datasets demonstrate the effectiveness of the proposed model in addressing challenges in sentiment analysis. The study utilizes Python programming language and various libraries, including Keras and Pandas.

Moving into the domain of financial sentiment analysis, Ioannis Almalis, Eleftherios Kouloumpris, and Ioannis Vlahavas[3] present a deep-learning solution in their research. Titled "Deep Learning Solution for Financial Sentiment Analysis," the authors address the limitations of traditional NLP approaches. Their model, exclusively trained on financial news, employs multiple recurrent neural networks and incorporates semi-supervised learning for detecting and correcting mislabeled data. The study achieves competitive results against state-of-the-art methods and introduces a novel sector-level sentiment analysis system. This research holds promise for predicting economic sectors affected by news articles, offering potential applications in sector fund indices.

In a related domain, Hugo Queiroz Abonizio, Emerson Cabrera Paraiso, and Sylvio Barbon, Jr.[4] explore text data augmentation methods for sentiment analysis. Their work delves into issues like imbalanced and limited datasets, proposing solutions to enhance classifier performance. The authors investigate augmentation techniques such as easy data augmentation, back-translation, BART, and pre-trained data augmentor, assessing their impact on sentiment analysis classifiers like LSTM, CNN, BERT, SVM, GRUs, RF, and ERNIE. The study covers scenarios of imbalanced and limited data, revealing that augmentation methods improve performance, particularly for reduced datasets. The article contributes a taxonomy of NLP augmentation methods, offering insights into the promising combination of augmentation with various classifiers.

Lastly, Dong Deng, Liping Jing, Jian Yu, and Shaolong Sun[5] introduce a novel Sparse Self-Attention LSTM (SSALSTM) for sentiment lexicon construction. In their paper, "Unsupervised Semantic Approach of ABSA for Large-Scale User Reviews," the authors propose a unique self-attention mechanism that considers the importance of each word in distinguishing document sentiment polarities. The introduced L1 regularization ensures sparsity, emphasizing only a minority of words in sentiment classification. Experiments on SemEval 2013–2016 datasets demonstrate that SSALSTM achieves state-of-the-art performance in both supervised and unsupervised sentiment classification tasks. This work provides an effective solution for sentiment lexicon construction in the era of deep learning.

III. METHODOLOGY

1. **Data Collection:** Gathering textual data from various sources, such as social media, reviews, or any other text-based content.

2. Text Preprocessing:

a) Tokenization: Break the textual data into individual words or tokens. Tokenization is the action of dividing a text into smaller units, known as tokens. These tokens can be words, phrases, symbols, or other meaningful elements, depending on the specific requirements of natural language processing (NLP) tasks. Tokenization is an essential step in text processing and analysis, enabling computers to understand and analyze textual data more effectively. It serves as the basis for various NLP applications such as text classification, named entity recognition, and part-of-speech tagging. Tokenization facilitates the extraction of meaningful information from raw text and forms the essential component for subsequent linguistic analysis and machine-learning tasks.

b) Lowercasing: Convert all text to lowercase to ensure uniformity. Lowercasing is the action of converting all letters in a text to lowercase. This transformation is commonly applied to normalize the text and ensure consistency, especially in natural language processing (NLP) tasks and text analysis. Lowercasing helps eliminate variations in the case of

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, February 2024

letters, treating uppercase and lowercase versions of the same word as identical. This ensures that the words are represented consistently, making it easier for algorithms to analyze and process the text without being sensitive to the letter case. Lowercasing is often a preprocessing step in tasks such as text classification, sentiment analysis, and information retrieval.

c) Removing Stopwords: Eliminate common words that don't contribute significant meaning. Removing stopwords involves eliminating common words that often do not contribute significant meaning to a text. These words, known as stopwords, are typically common in a language but may not carry substantial information in the context of certain natural language processing (NLP) tasks. Examples of stopwords include articles ("the," "a," "an"), prepositions ("in," "on," "at"), and conjunctions ("and," "but," "or").
The action of removing stopwords is a common pre-processing step in natural language processing (NLP) tasks such as text analysis, sentiment analysis, and information retrieval. By removing these words, the focus shifts to the more meaningful and contextually relevant terms, allowing algorithms to better capture the essence of the text and improve the efficiency of downstream tasks. This step aids in reducing noise and improving the overall quality of text analysis by retaining only words that carry more semantic weight.

d) Removing Punctuation: Exclude punctuation marks from the text. Removing punctuation involves eliminating any punctuation marks from a text. Punctuation marks include symbols such as periods, commas, question marks, exclamation points, quotation marks, parentheses, and others. The action of removing punctuation is often part of text preprocessing in natural language processing (NLP) tasks. The goal of removing punctuation is to simplify the text and focus on the essential words or tokens. Punctuation marks, while important for conveying grammatical structure and meaning, are often unnecessary in certain NLP applications, such as text classification, sentiment analysis, or word frequency analysis. Removing punctuation helps standardize the representation of words and facilitates subsequent analysis by algorithms, which can be sensitive to the presence of punctuation marks.

3. Feature Extraction:
a) Bag-of-Words (BoW): Represent the text as a vector of word frequencies.
The Bag of Words (BoW) model, without referring to specific algorithms or models, is a text representation technique used in NLP. This model assumes that the order of words is not considered, focusing solely on the presence or absence of words and their frequencies. It simplifies text representation for various NLP tasks, making it suitable for machine learning applications like text classification, sentiment analysis, and information retrieval. While lacking in capturing sequential information, BoW serves as a foundational concept for more advanced text representation models in NLP.

b) Word2Vec:Word embeddings using neural networks.
Word2Vec, a neural network-based feature extraction technique in natural language processing, is pivotal in sentiment analysis. Trained on extensive text data, it represents words as high-dimensional vectors, capturing distinct features. In the context of laptop review sentiment analysis, it transforms words into feature vectors, allowing representation of reviews as numerical vectors. This approach effectively captures underlying word meanings, facilitating sentiment prediction using machine learning models like logistic regression or neural networks. Word2Vec's ability to represent text as numerical vectors enhances its utility in sentiment analysis tasks.

c) TF-IDF (Term Frequency-Inverse Document Frequency): Reflects the importance of words in the context of the entire dataset.
TF-IDF, or Term Frequency-Inverse Document Frequency, is a numerical statistic used in natural language processing (NLP) and information retrieval to evaluate the importance of a word in a document relative to a collection of documents (corpus). Here's a brief explanation of TF-IDF:
(i). Term Frequency (TF):
- Measures how often a term (word) appears in a document.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, February 2024

- It is calculated as the ratio of the number of times a term occurs in a document to the total number of words in that document.
- Formula: \(TF(t, d) = \frac{\text{Number of times term t appears in document d}}{\text{Total number of terms in document d}}\)

(ii). Inverse Document Frequency (IDF):
- Measures the importance of a term across the entire corpus.
- It is calculated as the logarithm of the ratio of the total number of documents in the corpus to the number of documents containing the term, with 1 added to prevent division by zero.
-Formula: \(IDF(t, D) = \log\left(\frac{\text{Total number of documents in the corpus D}}{\text{Number of documents containing term t}}\right) + 1\)

(iii). TF-IDF Score:
- Combines TF and IDF to produce a weight for each term in a specific document.
- The TF-IDF score of a term in a document is the product of its TF and IDF scores.
- Formula: \(TFIDF(t, d, D) = TF(t, d) \times IDF(t, D)\)

The TF-IDF score reflects both the local importance of a term in a document and its global importance in the entire corpus. Higher TF-IDF scores indicate that a term is more important to a specific document relative to the entire corpus. TF-IDF is commonly used in text mining, information retrieval, and document classification tasks.

4. Model Selection:

a) Traditional ML Models: Models like Random Forests, Support Vector Machines (SVM),orNaive Bayes can be used for sentiment analysis purposes.

b) Deep Learning Models: Utilize neural networks, such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), or Transformer-based models like BERT can also be used.

i) Naïve Bayes:

Classification was performed using the Naive Bayes algorithm. Functioning as a probabilistic classifier that utilizes conditional probability to ascertain the likelihood of input belonging to a particular class [32]. Equation 9 outlines the calculation of probabilities for each involved class, incorporating conditional probability, prior probability, and evidence. Thomas Bayes refers the output as the posterior probability, expressed mathematically as:

\[P(y | X) = \frac{P(X | y) \cdot P(y)}{P(X)} \] (Equation 10)

In this equation:
- \(P(y) \): Prior Probability
- \(P(X | y) \): Likelihood probability
- \(P(y | X) \): Posterior Probability
- \(P(X) \): Marginal probability (Evidence)

Interpreting Equation 10 implies the occurrence of event \(y \) given that event \(X \) has occurred. Here, \(y \) represents the hypothesis, and \(X \) signifies the evidence, with the event \(y \) being dependent on the occurrence of event \(X \). \(X \) comprises independent features (i.e., \(x_1, x_2, \ldots, x_n \)). Various types of Naive Bayes exist, but our study specifically applied multinomial Naive Bayes, a model tailored for document processing and classification problems [33].

The multinomial Naive Bayes model assumes that each feature follows a multinomial distribution, where each word contributes to class prediction. Equation 11 defines the multinomial Naive Bayes as:

\[P(c | d) = \frac{P(c) \prod_{i=1}^{n} P(w_i | c)^{f_i}}{P(d)} \] (Equation 11)

In this equation:
- \(f_i \) is the frequency of a word \(w_i \) in document \(d \).
- \(P(c) \) is the prior probability that class \(c \) belongs to the document.
- \(P(w_i | c) \) is the conditional probability that the word occurs in the document belonging to class \(c \).

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, February 2024

ii) Support Vector Machine (SVM)

The Support Vector Machine (SVM) is acknowledged as a robust model capable of effectively classifying data into predefined categories [34]. This model establishes decision boundaries, referred to as hyperplanes, to delineate classes. Three distinct hyperplanes are identified: the positive hyperplane (Equation 12), the negative hyperplane (Equation 13), and the optimal hyperplane (Equation 14).

$$\mathbf{w} \cdot \mathbf{x} + b = 1 \text{ for Positive hyperplane} \quad \rightarrow \text{equation 12}$$

$$\mathbf{w} \cdot \mathbf{x} + b = -1 \text{ for Negative hyperplane} \quad \rightarrow \text{equation 13}$$

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \text{ for Optimal hyperplane} \quad \rightarrow \text{equation 14}$$

Here, $\mathbf{w}$ represents the width of the margin, b is the bias, and $\mathbf{x}$ denotes the features. Maximizing the margin width is crucial for achieving the best optimal hyperplane. In scenarios where the problem is non-linear, SVM excels by utilizing kernels, facilitating the transformation of data into higher dimensions, thereby enhancing separability. SVM supports various kernels, including polynomial, Gaussian, and Gaussian radial basis functions (RBF).

iii) Random Forest

The Random Forest model was employed as another classification tool, known for its ensemble machine-learning approach. This algorithm constructs multiple decision trees for classification, where the majority vote of these trees determines the class category. The methodology involves randomization and aggregation of tree predictions to produce the final output.

For Random Forest to function effectively, at least three hyperparameters must be specified: node size, the number of trees, and the number of features sampled. The algorithm utilizes bootstrap aggregation, commonly referred to as the bagging ensemble technique. This technique involves creating diverse subsets of training data derived from the original sample training set.

iv) Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) are specialized for sequential data processing, featuring a hidden state for context retention. They perform well in tasks such as language modelling and time series analysis but encounter challenges like vanishing/exploding gradients. Architectural variations like LSTMs and GRUs address these issues, sharing parameters across time steps. RNNs, however, struggle with capturing long-range dependencies. Bidirectional RNNs process sequences bidirectionally. Despite their effectiveness, RNNs have limitations, prompting the adoption of alternative architectures like transformers for specific applications.

v) Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) is a specific type of recurrent neural network (RNN) architecture designed to address the complexities of learning and capturing long-term dependencies in sequential data. LSTMs utilize memory cells and an advanced gating mechanism to selectively retain or discard information across multiple time steps. This functionality allows them to effectively capture and remember crucial patterns, making them particularly well-suited for tasks involving temporal dynamics. LSTMs have gained popularity in various applications, including natural language processing, speech recognition, and time series prediction. Their distinctive feature of mitigating the vanishing gradient problem sets them apart from traditional RNNs, enhancing their ability to model intricate sequences.

vi) BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a state-of-the-art natural language processing (NLP) model developed by Google. It was introduced in the paper titled "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," written by Jacob Devlin and his colleagues at Google Research.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, February 2024

Key features of the BERT model:
- Bidirectional Context:BERT is a model designed to capture contextual information bidirectionally, considering both the left and right context of each word in a sentence. This bidirectional approach allows the model to understand the meaning of words in the context of the entire sentence.
- Transformer Architecture: BERT is built on the Transformer architecture, which includes self-attention mechanisms. Transformers enable efficient processing of sequential data by considering dependencies between different parts of the input.
- Pre-training on Large Corpora:BERT is pre-trained on a large amount of unlabelled text data, learning contextualized representations of words. This pre-training phase is crucial for capturing general language understanding.
- Fine-tuning for Specific Tasks:After pre-training, BERT can be fine-tuned on smaller labelled datasets for specific NLP tasks, such as text classification, named entity recognition, and question answering.
- Masked Language Model (MLM):BERT uses a masked language model during pre-training. Some words in the input sequence are randomly masked, and the model learns to predict these masked words based on the context provided by the surrounding words.
- Next Sentence Prediction (NSP):BERT is trained to predict whether a randomly selected sentence follows another sentence in a document. This helps the model understand the relationships between sentences.
BERT has achieved remarkable performance on various NLP benchmarks and tasks, outperforming previous models in tasks that require a deep understanding of context and semantics. Its pre-trained representations can be leveraged for a wide range of downstream applications, making it a widely adopted and influential model in the field of natural language processing.

5. Training the Model:
Supervised Learning: Train the model on labelled data, where sentiments are already annotated.

6. Evaluation:
-Evaluate the model's performance on a distinct dataset that was not part of the training process, known as testing data.
-Utilize metrics such as accuracy, precision, recall, and F1 score to assess the effectiveness of the model.

7. Paraphrasing Steps:
1. Understand the Text:
-Read and Comprehend: Gain a clear understanding of the original text's meaning and sentiment.
2. Identify Key Sentences/Phrases:
- Select Important Content: Highlight crucial sentences or phrases that convey the main sentiment.
3. Restructure Sentences:
- Reword Sentences: Replace words with synonyms, rephrase, or change the sentence structure while retaining the original meaning.
-Maintain Sentiment: Ensure the sentiment conveyed in the paraphrased text aligns with the sentiment in the original.
4. Review and Refine:
- Quality Check: Ensure the paraphrased text is coherent, grammatically correct, and effectively conveys the sentiment.
5. Iterate if Necessary:
- Fine-tune: Make adjustments as needed, refining the paraphrased content for improved clarity and coherence.
6. Finalize Paraphrased Text:
- Create a Polished Version: Produce a final version of the paraphrased text that effectively conveys the sentiment while avoiding duplication.
Both sentiment analysis and paraphrasing play crucial roles in understanding and presenting text data in a meaningful way, whether for insights into opinions or for creating diverse textual representations.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, February 2024

IV. CONCLUSION

The fusion of machine learning techniques, optimization algorithms, and deep learning, as exemplified by the Deep-Recurrent Neural Network (D-RNN) introduced in this study, enhances sentiment analysis precision across various datasets. The application of sentiment analysis, a vital facet of natural language processing (NLP), extends to diverse domains such as social networking, market research, and opinion analysis. Optimization methods like gradient descent, genetic algorithms, Particle Swarm Optimization (PSO), and others play a crucial role in refining machine learning models, ensuring they capture the nuances of human emotions accurately. As businesses increasingly rely on sentiment analysis to understand user behaviour and opinions, the presented model and its integrated methodologies pave the way for improved accuracy and efficiency in this critical NLP application.

REFERENCES

[1]. H. Liu, X. Chen and X. Liu, "A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis," in IEEE Access, vol. 10, pp. 32280-32289, 2022, doi: 10.1109/ACCESS.2022.3160172.

[2]. P. Durga and D. Godavarthi, "Deep-Sentiment: An Effective Deep Sentiment Analysis Using a Decision-Based Recurrent Neural Network (D-RNN)," in IEEE Access, vol. 11, pp. 108433-108447, 2023, doi: 10.1109/ACCESS.2023.3320738.

[3]. Almalis, Ioannis, Eleftherios Kouloumpris, and Ioannis Vlahavas. "Sector-level sentiment analysis with deep learning." *Knowledge-Based Systems* 258 (2022): 109954.

[4]. H. Q. Abonizio, E. C. Paraiso and S. Barbon, "Toward Text Data Augmentation for Sentiment Analysis," in IEEE Transactions on Artificial Intelligence, vol. 3, no. 5, pp. 657-668, Oct. 2022, doi: 10.1109/TAI.2021.3114390.

[5]. D. Deng, L. Jing, J. Yu and S. Sun, "Sparse Self-Attention LSTM for Sentiment Lexicon Construction," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 27, no. 11, pp. 1777-1790, Nov. 2019, doi: 10.1109/TASLP.2019.2933326.

[6]. J. Yu, K. Chen and R. Xia, "Hierarchical Interactive Multimodal Transformer for Aspect-Based Multimodal Sentiment Analysis," in IEEE Transactions on Affective Computing, vol. 14, no. 3, pp. 1966-1978, 1 July-Sept. 2023, doi: 10.1109/TAFFC.2022.3171091.

[7]. M. Huang, H. Xie, Y. Rao, Y. Liu, L. K. M. Poon and F. L. Wang, "Lexicon-Based Sentiment Convolutional Neural Networks for Online Review Analysis," in IEEE Transactions on Affective Computing, vol. 13, no. 3, pp. 1337-1348, 1 July-Sept. 2022, doi: 10.1109/TAFFC.2020.2997769.

[8]. Z. Ren, G. Zeng, L. Chen, Q. Zhang, C. Zhang and D. Pan, "A Lexicon-Enhanced Attention Network for Aspect-Level Sentiment Analysis," in IEEE Access, vol. 8, pp. 93464-93471, 2020, doi: 10.1109/ACCESS.2020.2995211.

[9]. Y. -C. Tsai and F. -C. Lin, "Paraphrase Generation Model Integrating Transformer Architecture, Part-of-Speech Features, and Pointer Generator Network," in IEEE Access, vol. 11, pp. 30109-30117, 2023, doi: 10.1109/ACCESS.2023.3260849.

[10]. D. Zeng, H. Zhang, L. Xiang, J. Wang and G. Ji, "User-Oriented Paraphrase Generation With Keywords Controlled Network," in IEEE Access, vol. 7, pp. 80542-80551, 2019, doi: 10.1109/ACCESS.2019.2923057.

[11]. Y. Dong, Y. Fu, L. Wang, Y. Chen, Y. Dong and J. Li, "A Sentiment Analysis Method of Capsule Network Based on BiLSTM," in IEEE Access, vol. 8, pp. 37014-37020, 2020, doi: 10.1109/ACCESS.2020.2973711.

[12]. Punetha, Neha, and Goonjan Jain. "Bayesian game model based unsupervised sentiment analysis of product reviews." *Expert Systems with Applications* 214 (2023): 119128.

[13]. M. E. Basiri et al., "Improving Sentiment Polarity Detection Through Target Identification," in IEEE Transactions on Computational Social Systems, vol. 7, no. 1, pp. 113-128, Feb. 2020, doi: 10.1109/TCSS.2019.2951326.

[14]. S. M. Al-Ghuribi, S. A. Mohd Noah and S. Tiun, "Unsupervised Semantic Approach of Aspect-Based Sentiment Analysis for Large-Scale User Reviews," in IEEE Access, vol. 8, pp. 218592-218613, 2020, doi: 10.1109/ACCESS.2020.3042312.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 1, February 2024

[15]. P. Vyas, M. Reisslein, B. P. Rimal, G. Vyas, G. P. Basyal and P. Muzumdar, "Automated Classification of Societal Sentiments on Twitter With Machine Learning," in IEEE Transactions on Technology and Society, vol. 3, no. 2, pp. 100-110, June 2022, doi: 10.1109/TTS.2021.3108963.

[16]. L. Huang, Z. Dou, Y. Hu and R. Huang, "Online Sales Prediction: An Analysis With Dependency SCOR-Topic Sentiment Model," in IEEE Access, vol. 7, pp. 79791-79797, 2019, doi: 10.1109/ACCESS.2019.2919734.

[17]. X. Li, F. Zeng and C. Yao, "A Semi-Supervised Paraphrase Identification Model Based on Multi-Granularity Interaction Reasoning," in IEEE Access, vol. 8, pp. 60790-60800, 2020, doi: 10.1109/ACCESS.2020.2984009.

[18]. J. Liu et al., "Noun Compound Interpretation With Relation Classification and Paraphrasing," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 9, pp. 8757-8769, 1 Sept. 2023, doi: 10.1109/TKDE.2022.3208617.

[19]. K. Mrinalini, P. Vijayalakshmi and T. Nagarajan, "SBSim: A Sentence-BERT Similarity-Based Evaluation Metric for Indian Language Neural Machine Translation Systems," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 30, pp. 1396-1406, 2022, doi: 10.1109/TASLP.2022.3161160.

[20]. J. Yu, K. Chen and R. Xia, "Hierarchical Interactive Multimodal Transformer for Aspect-Based Multimodal Sentiment Analysis," in IEEE Transactions on Affective Computing, vol. 14, no. 3, pp. 1966-1978, 1 July-Sept. 2023, doi: 10.1109/TAFFC.2022.3171091.

[21]. Zhang, Lumin, et al. "User-level sentiment evolution analysis in microblog." *China Communications* 11.12 (2014): 152-163.hn

[22]. N. Li, C. -Y. Chow and J. -D. Zhang, "SEML: A Semi-Supervised Multi-Task Learning Framework for Aspect-Based Sentiment Analysis," in IEEE Access, vol. 8, pp. 189287-189297, 2020, doi: 10.1109/ACCESS.2020.3031665.

[23]. J. Z. Maitama, N. Idris, A. Abdi, L. Shuib and R. Fauzi, "A Systematic Review on Implicit and Explicit Aspect Extraction in Sentiment Analysis," in IEEE Access, vol. 8, pp. 194166-194191, 2020, doi: 10.1109/ACCESS.2020.3031217.

[24]. L. -C. Yu, J. Wang, K. R. Lai and X. Zhang, "Refining Word Embeddings Using Intensity Scores for Sentiment Analysis," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 3, pp. 671-681, March 2018, doi: 10.1109/TASLP.2017.2788182.

[25]. S. Rida-E-Fatima et al., "A Multi-Layer Dual Attention Deep Learning Model With Refined Word Embeddings for Aspect-Based Sentiment Analysis," in IEEE Access, vol. 7, pp. 114795-114807, 2019, doi: 10.1109/ACCESS.2019.2927281.

[26]. J. He, A. Wumaier, Z. Kadeer, W. Sun, X. Xin and L. Zheng, "A Local and Global Context Focus Multilingual Learning Model for Aspect-Based Sentiment Analysis," in IEEE Access, vol. 10, pp. 84135-84146, 2022, doi: 10.1109/ACCESS.2022.3197218.

[27]. K. R. Mabokela, T. Celik and M. Raborife, "Multilingual Sentiment Analysis for Under-Resourced Languages: A Systematic Review of the Landscape," in IEEE Access, vol. 11, pp. 15996-16020, 2023, doi: 10.1109/ACCESS.2022.3224136.

[28]. K. Chakraborty, S. Bhattacharyya and R. Bag, "A Survey of Sentiment Analysis from Social Media Data," in IEEE Transactions on Computational Social Systems, vol. 7, no. 2, pp. 450-464, April 2020, doi: 10.1109/TCSS.2019.2956957.

[29]. S. Bengesi, T. Oladunni, R. Olusegun and H. Audu, "A Machine Learning-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion From Twitter Tweets," in IEEE Access, vol. 11, pp. 11811-11826, 2023, doi: 10.1109/ACCESS.2023.3242290.

[30]. Kalamatianos, Georgios, et al. "Towards the creation of an emotion lexicon for microblogging." *Journal of Systems and Information Technology* 20.2 (2018): 130-151.