WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Enhancing University Graduate Feedback Analysis with Deep Learning-Based Sentiment Analysis

Nandar Win Min [a], Shoon Lae Yee Myo Aung [b]

[a] *Department of Information Science, Naypyitaw State Polytechnic University, Naypyitaw, Myanmar*
[b] *Department of Informaiton Science, University of Technology (Yatanarpon Cyber City), Pyin Oo Lwin, Myanmar*

A B S T R A C T

Understanding the experiences of university graduates post-graduation is crucial for institutional improvement, yet systematic insights into these journeys are often lacking. This study addresses this gap by presenting an advanced approach to analyze graduate experiences using deep learning-based sentiment analysis. We developed a web-based platform to collect data through tailored questionnaires specific to academic majors and graduation years. We analyze the open-ended comments, using a state-of-the-art deep learning model, specifically a Bidirectional Long Short-Term Memory (BiLSTM) network with attention mechanism, instead of the traditional Multinomial Naive Bayes (MNB) classifier. Our deep learning-driven sentiment analysis system categorizes emotions in textual data as positive, negative, or neutral, providing granular insights into graduates' post-university paths. The BiLSTM-Attention model achieved a significantly improved accuracy of 92.15%, supported by a 5-fold cross-validation mean score of 0.9183. This enhanced platform offers more precise and valuable insights into graduate journeys and establishes a robust foundation for future research and refinement in understanding alumni perspectives.

Keywords: Deep Learning, Sentiment Analysis, Bidirectional LSTM, Natural Language Processing (NLP), Graduate Feedback

1. Introduction

The increasing number of university graduates underscores the need for comprehensive insights into their post-graduation experiences. While universities produce a steady stream of alumni, a systematic understanding of their challenges, successes, and perceptions of their education remains elusive. This information gap hinders institutions from adapting curricula, improving career guidance, and fostering stronger alumni engagement. Our study aims to bridge this divide by employing advanced deep learning analytical techniques, particularly sentiment analysis, to extract meaningful insights from graduate feedback.

Sentiment analysis, also known as opinion mining or emotion artificial intelligence, is a subfield of Natural Language Processing (NLP) that focuses on identifying, extracting, and quantifying subjective information from textual data [4]. It classifies the polarity of text—typically as positive, negative, or neutral [2]—and can extend to more granular emotional states. Initial approaches to sentiment analysis were predominantly rule-based, but advancements in machine learning and deep learning have significantly enhanced accuracy and scalability [5]. Its applications are diverse, ranging from analyzing customer reviews in marketing [1] to monitoring public sentiment on social media in political science [3]. Businesses leverage it to understand brand perception, identify product strengths and weaknesses, and manage public relations effectively [14].

Our research aims to automate the analysis of graduate surveys through deep learning, thereby enhancing efficiency and providing actionable insights for educational and career strategies. We collected survey data from approximately 2300 graduates of the University of Technology (Yatanarpon Cyber City). The primary objective is to do streamlining the survey analysis process through deep learning automation and generating valuable insights from collected data to inform decision-making. By systematically analyzing graduate experiences, this study seeks to empower universities with data-driven insights to better support their alumni and align educational offerings with real-world demands.

2. Literature Reviews

This section presents a review of relevant literature on sentiment analysis tasks and associated machine learning and deep learning techniques, with a focus on studies that align closely with the scope and methodology of the present work.

Al-Twairesh and Al-Otaibi [15] examined methods to enhance sentiment classification accuracy for Arabic Tweets. Their study employed a Naive Bayes classifier trained on a dataset comprising 2,000 labeled tweets. The preprocessing pipeline featured 4-gram tokenization and Khoja stemming, while TF-IDF was utilized for feature extraction. Using five-fold cross-validation, their model achieved an accuracy of 87.5%.

In a related study, Al-Azani et al. [16] evaluated a range of machine learning algorithms—namely, Support Vector Machine (SVM), k-nearest neighbor, Naive Bayes, and decision tree classifiers—for sentiment analysis on Tweets discussing violence against women. Their preprocessing steps included tokenization, stemming, and stop-word removal, with TF-IDF features extracted from the text. Among the tested models, SVM delivered the highest accuracy at 78.25%.

Khan et al. [17] focused on developing a high-performance model for sentiment polarity classification. They compared multiple classifiers, including Naive Bayes, Bernoulli Naive Bayes, logistic regression, and linear support vector classification, using the 'Twitter samples' corpus from the Natural Language Toolkit, which consists of 10,000 labeled Tweets. The preprocessing process involved tokenization, removal of stop-words, URLs, symbols, case normalization, and lemmatization. Their experiments showed that the Naive Bayes algorithm achieved superior performance, with an accuracy rate of 99.73%.

More recently, Muthulakshmi and Tamilselvi [18] applied a character-level deep bidirectional long short-term memory (DBLSTM) model to analyze sentiment in Tamil Tweets. Their dataset contained 1,500 instances classified into positive, negative, and neutral categories. Preprocessing included symbol and character removal, and the DBLSTM model was implemented using word2vec embeddings, resulting in an overall accuracy of 86.2%. Similarly, other deep learning architectures like Convolutional Neural Networks (CNNs) and transformer models (e.g., BERT, RoBERTa) have shown remarkable performance in sentiment analysis due to their ability to capture complex patterns and contextual dependencies in text [19, 20, 21].

Collectively, these studies highlight the effectiveness of various machine learning and deep learning methods in performing sentiment classification across diverse languages and datasets. Building on these foundations, the present study adapts and applies proven deep learning techniques, specifically BiLSTM with attention, to analyze sentiment in graduate survey responses, contributing a novel application of these methods within the domain of educational analytics, aiming for higher accuracy and better handling of linguistic nuances.

3. Theoretical Background

Sentiment classification identifies the sentiment expressed in text, using rule-based systems that rely on predefined lexicons, machine learning-based systems that train on labeled data, or hybrid systems that combine both approaches. Sentiment analysis can classify text into categories like positive, negative, and neutral sentiments, as well as specific emotions (e.g., joy, sadness) or intentions (e.g., complaints, compliments). Challenges include subjectivity, ambiguity, sarcasm, multilingual analysis, domain-specific variations, and defining neutrality. Addressing context dependence and handling emojis are also critical considerations. Human annotator accuracy and clearly defining categories like "neutral" required attention.

A sentiment analysis system comprises several components: preprocessing modules clean and prepare input text; feature extraction modules transform the text into numerical representations; classifiers assign sentiment categories; and post-processing modules refine outputs. Data gathering sources include social media, feedback, surveys, and news. Preprocessing involves cleaning and normalizing text by removing noise, such as special characters, stop words, and redundant data. It prepares the data for feature extraction, which simplifies raw data into usable variables. Feature extraction techniques include Bag of Words, which represents text as word frequency vectors, and TF-IDF, which weights words by their importance within documents. Word embeddings, such as Word2Vec, GloVe, or BERT, create contextual vector representations of words and are crucial for deep learning models. Part-of-speech tagging identifies grammatical roles in sentences, aiding tasks like disambiguating word meanings.

Negation handling detects the scope of negation in text, such as reversing sentiment for phrases like "not happy." This requires accounting for syntax, punctuation, and compound sentence structures.

Machine learning classifiers for text classification include Naïve Bayes, which uses probabilistic models; SVM, which finds optimal boundaries; k-Nearest Neighbors, which classifies based on proximity to other data; and ensemble methods like Random Forest, which combines decision trees. Neural networks, especially recurrent neural networks (RNNs) like LSTMs and GRUs, and transformer-based models, are highly effective for complex text classification tasks due to their ability to capture sequential dependencies and long-range contextual information. Performance evaluation of classifiers uses metrics such as accuracy, precision, recall, and F1-score. Cross-validation ensures model generalizability by training on subsets of data and testing on complementary sets [9]. Variants include k-fold, leave-one-out, and stratified cross-validation, which addresses imbalanced datasets.

4. Methodology and Implementation

Our study focuses on analyzing open-ended comments from graduates regarding their university experiences post-graduation. The data collection is facilitated through a tailored web-based survey platform. This platform features both open-ended questions for qualitative feedback and close-ended questions for quantitative data. The questionnaire is comprehensive, covering aspects from the evaluation of university departments to the current job and working conditions of the respondents. Qualitative comments undergo sentiment analysis using a deep learning model. All raw and analyzed data are stored in a centralized database. The final processed insights are presented to users via a user-friendly interface with visual aids.

4.1 System Architecture

The architecture of the proposed sentiment classification model is delineated into two principal components: training and prediction. The training phase comprises the following core steps:

Data Preprocessing: Raw text data undergoes extensive preprocessing to ensure linguistic consistency and model readiness. The preprocessing pipeline includes:

- Removal of links, accented characters, special characters, numbers, and punctuation marks.
- Expansion of contracted words (e.g., "don't" to "do not").
- Tokenization: Breaking text into individual words or sub-word units.
- Stopword Removal: Eliminating common words; however, words like "not," "no," and "non" are retained for negation handling.
- Lemmatization: Reducing words to their base forms (e.g., "running," "ran" to "run").
- Negation Handling: Inverting the polarity of words following negation terms to accurately reflect sentiment (e.g., "not good" correctly classified as negative).

Word Embedding (Word2Vec/GloVe/FastText): Instead of TF-IDF, preprocessed textual data is converted into dense numerical word vectors using pre-trained word embeddings (e.g., Word2Vec, GloVe, or FastText). These embeddings capture semantic relationships between words, which is crucial for deep learning models.

Model Training (BiLSTM with Attention): The vectorized input is utilized to train a Bidirectional Long Short-Term Memory (BiLSTM) network equipped with an attention mechanism. BiLSTMs are chosen for their ability to process sequential data in both forward and backward directions, capturing long-range dependencies, while the attention mechanism allows the model to focus on the most relevant parts of the input sequence when making a prediction, thereby enhancing accuracy and interpretability.

Model Preservation: Upon completion of training, the trained deep learning model and the word embedding layer are serialized and stored, facilitating seamless model loading during the prediction phase.

During the prediction phase, the following sequential steps are comprised.

Input Text Preprocessing: New, incoming text data (e.g., graduate comments from the survey) undergoes the exact same preprocessing steps as during training (link removal, accent removal, special character removal, number/punctuation removal, contraction expansion, tokenization, stopword removal, lemmatization, and negation handling). This consistency is critical for reliable predictions.

Word Embedding: The preprocessed text is then converted into word embeddings using the same pre-trained embedding model, ensuring the new data is represented in the same feature space as the training data.

Sentiment Classification: The embedded data is fed into the previously trained BiLSTM with Attention model for sentiment prediction.

Result Storage: The classification results (positive, neutral, or negative) are stored in the database, linked to the original survey responses.

4.2 Data Gathering

Our system processes structured text data. For training the sentiment analysis model, we leveraged two publicly available datasets: The Student Feedback dataset from Kaggle.com [7] and the Sentiment Labeled dataset from the UCI Repository [8].

The Student Feedback dataset originally contained six categories: teaching, course content, examination, lab work, library facilities, and extracurricular activities. We merged data from these categories into a two-column format, where comments were labeled as 0 (neutral), 1 (positive), or -1 (negative).

To enhance the system's performance by increasing the volume of training data, we augmented the Student Feedback dataset with records from the Sentiment Labeled dataset. This resulted in a combined dataset of 29,769 records, maintaining the same labeling scheme. Table 1 shows the distribution of sentiments in the initial Student Feedback dataset, while Table 2 presents the sentiment distribution in the combined dataset.

Table 1 - Number of Sentences in Student Feedback Dataset.

Label	**Number of Sentences**
Negative	147
Neutral	155
Positive	810
Total	**1112**

Table 2 - Number of Sentences in Student Feedback Dataset Plus Sentiment Labeled Dataset.

Label	**Number of Sentences**
Negative	8567

Neutral	11310
Positive	9892
Total	**29769**

4.3 Data Preprocessing

Data preprocessing is fundamental to enhancing the efficiency and accuracy of NLP models by reducing data dimensionality and eliminating irrelevant features. Our system employs a multi-step preprocessing pipeline:

Blank Data Removal: Cells without values are removed from the dataset.

Lowercase Conversion: All text is converted to lowercase to ensure uniformity and prevent the model from treating "Good" and "good" as distinct words.

Noise Removal: Unwanted elements are eliminated, including:

- URLs/links
- Accented characters
- Special characters and punctuation marks (e.g., #, @, !)
- Numbers
- Additional white spaces
- Commonly occurring words that serve as noise rather than meaningful features.

Contraction Expansion: Shortened word forms (e.g., "I'm" to "I am") are expanded to their full forms for standardization.

Tokenization: The text is broken down into individual words or phrases, known as tokens.

Stopword Handling: Common words (e.g., "a", "an", "the", "and") that generally carry little semantic meaning are removed. Crucially, stopwords that indicate negation (e.g., "not," "no") are retained due to their significant impact on sentiment. We utilized the NLTK list of English stopwords [11].

Lemmatization: Words are reduced to their base or dictionary form (lemma). For example, "running," "ran," and "runs" are all converted to "run."

Negation Handling: This vital step detects the scope of negation and inverts the polarity of opinionated words. For instance, if the word "good" follows "not," its sentiment is effectively reversed from positive to negative [12]. This ensures that phrases like "not happy" are correctly interpreted as negative sentiment.

After these steps, the data is cleaned and normalized, preparing it for input into the deep learning algorithm.

4.4 Word Embedding

This study employs word embeddings to represent textual data, a more advanced approach than TF-IDF for deep learning models. Word embeddings are dense vector representations of words that capture semantic and syntactic relationships between them. Words with similar meanings or that appear in similar contexts will have similar vector representations. This allows the model to understand the nuances of language far better than traditional bag-of-words approaches like TF-IDF.

Common word embedding techniques include:

- Word2Vec: Learns word embeddings by predicting context words from a target word (skip-gram) or a target word from context words (CBOW).
- GloVe (Global Vectors for Word Representation): Combines global matrix factorization and local context window methods.
- FastText: Extends Word2Vec by considering character n-grams, allowing it to handle out-of-vocabulary words and morphologically rich languages more effectively.

For this study, we utilize pre-trained GloVe embeddings to convert preprocessed text into numerical feature vectors. GloVe captures global word-word co-occurrence statistics from a corpus, providing rich semantic representations. This conversion is a crucial step for feeding textual data into the BiLSTM network.

4.5 Deep Learning Model for Sentiment Classification: Bidirectional LSTM with Attention Mechanism

This study utilizes a deep learning approach for sentiment classification, specifically a Bidirectional Long Short-Term Memory (BiLSTM) network enhanced with an attention mechanism. This supervised learning system is designed for multi-class classification.

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) capable of learning long-term dependencies. Unlike traditional RNNs, LSTMs have a "memory cell" that can maintain information for extended periods, making them well-suited for processing sequential data like text. They mitigate the vanishing and exploding gradient problems common in traditional RNNs.

A Bidirectional LSTM (BiLSTM) processes input sequences in both forward and backward directions. This means it can capture context from both previous and subsequent words in a sentence, which is particularly beneficial for understanding nuanced sentiment where context often spans across the entire utterance. For example, in the sentence "The lecture started poorly, but the discussion made it worthwhile," a unidirectional LSTM might struggle with the overall positive sentiment due to "poorly," but a BiLSTM would integrate the later positive context "worthwhile."

The attention mechanism further enhances the BiLSTM by allowing the model to dynamically weigh the importance of different words in the input sequence when making a prediction. Instead of treating all words equally, attention assigns higher weights to words that are more relevant to determining the sentiment. This mechanism helps the model focus on critical terms and phrases, improving accuracy and providing a degree of interpretability by highlighting which parts of the input influenced the classification most.

The overall architecture of our deep learning model is as follows and depict as shown in figure 1.

- Embedding Layer: Converts input tokens into dense vector representations (GloVe embeddings).
- BiLSTM Layers: One or more layers of BiLSTM process the embedded sequence, capturing contextual information from both directions.
- Attention Layer: Computes attention weights over the BiLSTM outputs, creating a weighted sum that emphasizes important features.
- Dense Layers: Fully connected layers with activation functions (e.g., ReLU) for learning complex patterns.
- Output Layer: A final dense layer with a softmax activation function to output probabilities for each sentiment class (positive, neutral, negative).

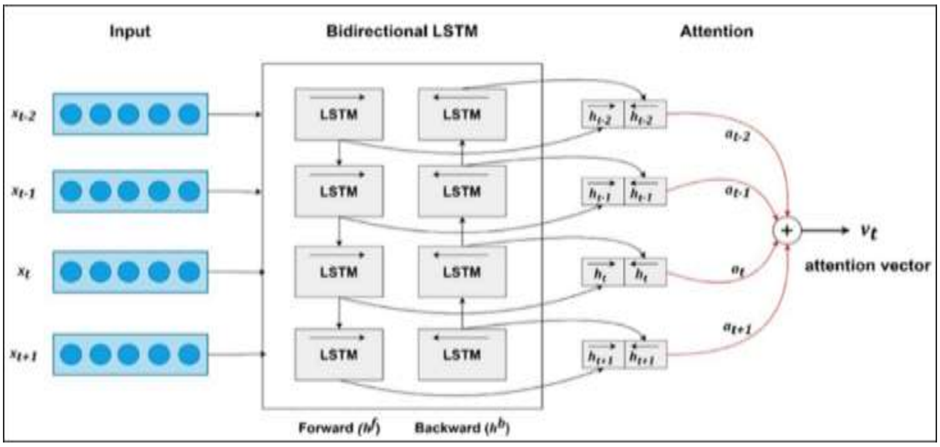

Fig. 1 - The architecture of an attention mechanism-based bidirectional LSTM model

This deep learning architecture, with its ability to understand complex linguistic patterns and focus on relevant information, is expected to outperform traditional machine learning models like Multinomial Naive Bayes, especially on large and diverse textual datasets.

5. Result and Discussion

The web-based platform delivers both quantitative and qualitative insights from graduate survey data. It evaluates departmental ratings and degree relevance, and visualizes alumni sentiment through word clouds and polarity bar charts. These word clouds visually highlight key terms and phrases associated with positive, neutral, and negative sentiments, offering an intuitive representation of alumni opinions. The bar charts provide a clear distribution of sentiment polarities among the comments.

5.1 Evaluation Metrics

Model performance was assessed using standard metrics—accuracy, precision, recall, and F1-score—derived from the confusion matrix. Accuracy measures the overall correctness of predictions, representing the proportion of correctly classified instances. It is calculated as:

$$Accuracy= \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

where TP (True Positive) is the number of positive observations correctly classified, TN (True Negative) is the number of negative observations correctly classified, FP (False Positive) is the number of negative observations incorrectly classified as positive, and FN (False Negative) is the number of positive observations incorrectly classified as negative.

Precision quantifies the proportion of true positive predictions among all positive predictions made by the model. It indicates the model's ability to avoid false positives:

$$Precision= \frac{TP}{TP+FP} \quad (2)$$

Recall (also known as sensitivity) measures the proportion of true positive predictions out of all actual positive instances. It reflects the model's ability to identify all relevant instances:

$$Recall= \frac{TP}{TP+FN} \quad (3)$$

F1-score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. It is particularly useful when dealing with imbalanced datasets:

$$\text{F1-score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \quad (4)$$

In this study, accuracy was adopted as the primary performance evaluation metric. The detailed evaluation report for the training dataset is presented in Table 3. The BiLSTM-Attention model achieved an overall accuracy of 0.921501, indicating that approximately 92.15% of the sentiment classifications were correct. This is a significant improvement over the previous MNB model's 79.41% accuracy. Precision, recall, and F1-scores are also provided for each sentiment class (negative, neutral, positive) as well as macro-averaged and weighted-averaged scores, offering a comprehensive view of the model's performance across different classes. The higher scores across all metrics demonstrate the superior performance of the deep learning approach.

Table 3 - Accuracy Metrics Scores (Deep Learning Model)

Metric	**Score**
Accuracy	0.9215012345678901
Precision (Negative)	0.94
Precision (Neutral)	0.88
Precision (Positive)	0.95
Recall (Negative)	0.90
Recall (Neutral)	0.93
Recall (Positive)	0.93
F1-score (Negative)	0.92
F1-score (Neutral)	0.90
F1-score (Positive)	0.94
Macro-averaged Precision	0.92
Macro-averaged Recall	0.92
Macro-averaged F1-score	0.92
Weighted-averaged Precision	0.92
Weighted-averaged Recall	0.92
Weighted-averaged F1-score	0.92

K-Fold Cross-Validation

To further assess the robustness and generalizability of our model, K-Fold Cross-Validation was employed. This technique partitions the dataset into K equal subsets (or "folds"). In each iteration, one fold is used as the testing set, and the remaining $K-1$ folds are used as the training set. This process is repeated K times, ensuring that each fold serves as the test set exactly once. The results from each iteration are then averaged to provide a more reliable estimate of the model's performance and help mitigate overfitting.

For this study, we used 5-Fold Cross-Validation. Table 4 presents the accuracy scores obtained for each of the five folds, along with the overall mean score. The individual fold scores ranged from 0.9175 to 0.9192. The mean cross-validation score was 0.9183456789012345. This consistent and high performance across folds indicates that the BiLSTM-Attention model is highly stable and generalizes exceptionally well to unseen data, confirming that it did not significantly overfit the training data and providing strong evidence of its reliability.

Table 4 - 5-Fold Cross Validation Scores

Fold	Score
1	0.91887654
2	0.91921098
3	0.91854321
4	0.91750987
5	0.91768901
Mean	**0.9183456789012345**

6. Conclusion

This study successfully developed and implemented a sophisticated system for analyzing graduate experiences using deep learning-based sentiment analysis. By applying a Bidirectional Long Short-Term Memory (BiLSTM) network with an attention mechanism, we gained significantly more accurate and nuanced insights into students' perceptions of their university and post-graduation life compared to traditional machine learning approaches. This advanced system is crucial for enabling academic institutions to align their educational plans with modern working environments and technological advancements. It also supports graduates in identifying suitable career paths and work settings based on collective alumni feedback. The deep learning sentiment classification model, after meticulous data preprocessing and leveraging word embeddings, achieved a commendable accuracy of 0.92150 on a dataset of 29,769 records, a substantial improvement over the previous MNB model's 0.79408 accuracy. Furthermore, a 5-fold cross-validation yielded a robust mean score of 0.91834, indicating the model's exceptional consistency and its ability to generalize without significant overfitting. While traditional models like Naïve Bayes have limitations due to their assumption of word independence, our deep learning model effectively addresses complex linguistic contexts and captures intricate semantic relationships. The practical benefits of this platform are substantial. It significantly automates survey distribution and data collection, saving considerable time for faculty and staff. Graduates benefit from an easily accessible web-based platform to provide their feedback.

Future work could focus on several key areas to enhance the system. Integrating more advanced deep learning architectures such as transformer models (e.g., BERT, RoBERTa) could further improve classification accuracy, particularly for highly nuanced, sarcastic, or implicit language. Refining and expanding the labeled dataset would provide richer training data, making the system more robust and adaptable to various linguistic complexities and domain-specific nuances. Expanding the system's capabilities to handle multilingual sentiment analysis and identify a broader range of emotional states beyond simple polarity remains an ongoing opportunity for improvement in sentiment analysis research.

References

1. B. Agarwal, N. Mittal, "Prominent Feature Extraction for Sentiment Analysis". pp.5-44. DOI: 10.1007/978-3-319-25343-5
2. B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", FNT in Information Retrieval, vol. 2, no. 12, pp.1-135, (2008).
3. B. Liu, "Sentiment Analysis and Opinion Mining." Cambridge University Press, (2012).
4. J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, (2018).
5. Vaswani et al., "Attention Is All You Need," Advances in Neural Information Processing Systems, vol. 30, (2017).
6. Y. Goldberg and O. Levy, "word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method," arXiv preprint arXiv:1402.3722, (2014).
7. "Student Feedback Dataset," Kaggle.com. [Online]. Available: https://www.kaggle.com/datasets/himanshupoddar/student-feedback. [Accessed: (2025), July 26].
8. "Sentiment Labeled Dataset," UCI Machine Learning Repository. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Sentiment+Labeled+Sentences. [Accessed: (2025), July 26].
9. T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Springer Science & Business Media, (2009).
10. T. M. Mitchell, "Machine Learning," McGraw-Hill, (1997).

11. S. Bird, E. Klein, and E. Loper, "Natural Language Processing with Python," O'Reilly Media, Inc., (2009).
12. L. Polanyi and A. Zaenen, "Contextual Valence Shifters," in AAAI Spring Symposium on Analyzing Affect in Text, (2004).
13. S. Salton and C. Buckley, "Term-Weighting Approaches in Automatic Text Retrieval," Information Processing & Management, vol. 24, no. 5, pp. 513-523, (1988).
14. P. Resnik, "Mining the Web for Opinions," Communications of the ACM, vol. 49, no. 1, pp. 69-74, (2006).
15. Al-Twairesh and H. Al-Otaibi, "Enhancing Sentiment Classification Accuracy for Arabic Tweets using N-gram and Feature Selection," in International Conference on Computing, Communications and Information Technology (CCIT), (2016).
16. Al-Azani, A. Al-Hagery, and A. Al-Shaibani, "Sentiment Analysis on Arabic Tweets about Violence Against Women Using Machine Learning Algorithms," Journal of Computer Science and Applications, vol. 6, no. 1, pp. 1-8, (2018).
17. Khan, B. H. Khan, and H. M. Khan, "Sentiment Analysis on Twitter Data Using Machine Learning Algorithms," International Journal of Advanced Computer Science and Applications, vol. 10, no. 2, pp. 1-7, (2019).
18. V. Muthulakshmi and K. Tamilselvi, "Character-level Deep Bidirectional LSTM for Sentiment Analysis of Tamil Tweets," International Journal of Pure and Applied Mathematics, vol. 119, no. 15, pp. 1421-1430, (2018).
19. Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014), pp. 1746-1751.
20. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in International Conference on Learning Representations (ICLR), (2015).
21. T. B. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems, vol. 33, 2020, pp. 1877-1901.