B.Tech ProjectReport

(ECIR80)

on

SENTIMELD:

LEXIHYB BASED SENTIMENT ANALYTICS

by

ANKITA (12015054)

VIBHUTI JINDAL (12015109)

YASH SINGHAL (12015068)

Under the Supervision of

Dr. Rajender Kumar

Assistant Professor

Department of Electronics and communication Engineering

National Institute of Technology, Kurukshetra

Haryana-136119, India

(May 2024)

Certificate

We, hereby certify that the work which is being presented in this B.Tech Project (CSPE40) report entitled "***SENTIMELD: Lexihyb Based Sentiment Analytics***", in partial fulfillment of the requirements for the **Bachelor of Technology in Electronics and communication Engineering** is an authentic record of our own work carried out during a period from January, 2024 to May, 2024 under the supervision of **Dr. Rajender Kumar, Assistant Professor**, Electronics and communication Engineering Department.

The matter presented in this project report has not been submitted for the award of any otherdegree elsewhere.

Signature of Candidate

Ankita (12015054)

Vibhuti Jindal(12015109)

Yash Singhal (12015068)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date: 28.04.2024

Signature of Supervisor Faculty Mentor
Dr. Rajender Kumar
Asst. Prof.

Table of Contents

Abstract

Sentimeld: LexiHyb Based Sentiment Analysis is a comprehensive project focused on evaluating customer sentiments towards unlocked mobile phones through the analysis of reviews sourced from Amazon. The project aims to provide valuable insights for businesses in the mobile phone industry to enhance their brand reputation and competitiveness in the market.

The proposed architecture encompasses a multi-step process, beginning with data preprocessing to address various challenges such as null values, non-English text, spelling errors, and noise. The dataset consists of approximately 400,000 reviews, ensuring a robust and representative sample for analysis. Through extensive preprocessing, including tokenization, stopword removal, and stemming, the text data is transformed into a format suitable for analysis

The project adopts a hybrid approach combining lexicon-based analysis and machine learning-based classification to classify reviews into positive, negative, or neutral sentiments. This approach leverages the strengths of both methods, incorporating semantic and contextual information to enhance the accuracy and robustness of sentiment analysis. Machine learning algorithms such as support vector machines, logistic regression, and random forests are trained on labeled datasets to classify reviews based on their sentiment. The second proposed approach enhances the traditional LSTM model by integrating Word2Vec embeddings to enhance its performance in sentiment analysis tasks. Unlike the conventional LSTM model, which relies on randomly initialized weights to convert numerical sequences of words into word embeddings, this approach leverages pretrained Word2Vec embeddings for the embedding layer initialization.

Experimental results demonstrate the effectiveness of the hybrid approach, achieving an accuracy of over 94.85% in sentiment classification. The analysis reveals significant variations in sentiment across different mobile phone brands, providing valuable insights into customer preferences and perceptions. Brands like Apple and Samsung receive predominantly positive reviews, reflecting their strong brand reputation and customer loyalty, while lesser-known brands face challenges in garnering positive sentiment. Overall, SentimeLD: LexiHyb Based Sentiment Analysis serves as a comprehensive resource for businesses in the mobile phone industry, offering actionable insights into customer sentiments and preferences. By leveraging the findings of sentiment analysis, brands can tailor their products, services, and marketing strategies to better meet consumer needs and expectations, driving growth and innovation in the industry

1. Introduction

Sentimeld: LexiHyb Based Sentiment Analysis is a comprehensive endeavor aimed at evaluating customer sentiments regarding unlocked mobile phones through an in-depth analysis of reviews obtained from Amazon. The primary objective of the project is to furnish valuable insights for businesses operating within the mobile phone industry, empowering them to bolster their brand reputation and competitive standing within the market.

At the core of the project lies a meticulously designed architecture, which initiates with meticulous data preprocessing procedures tailored to address various challenges inherent in the dataset. Comprising approximately 400,000 reviews sourced from Amazon, the dataset ensures a robust and representative sample for analysis. Through rigorous preprocessing techniques encompassing tasks such as null value handling, language detection, spelling error correction, and noise reduction, the raw text data undergoes transformation into a standardized format amenable to analysis.

The project adopts a hybrid approach, marrying lexicon-based analysis with machine learning-based classification methodologies, thereby capitalizing on the strengths of each method to enrich the accuracy and reliability of sentiment analysis. Leveraging semantic and contextual cues, this hybrid approach endeavors to provide nuanced insights into the sentiments expressed within the reviews. Machine learning algorithms, including support vector machines, logistic regression, and random forests, are trained on labeled datasets to discern and classify reviews based on their sentiment. Additionally, a novel enhancement to the traditional LSTM model is proposed, integrating Word2Vec embeddings to augment its performance in sentiment analysis tasks.

Experimental findings unveil the efficacy of the hybrid approach, showcasing an impressive accuracy rate exceeding 94.85% in sentiment classification. Furthermore, the analysis unearths notable disparities in sentiment across various mobile phone brands, offering crucial insights into customer preferences and perceptions. Established brands such as Apple and Samsung tend to elicit predominantly positive reviews, indicative of their robust brand reputation and customer loyalty. Conversely, lesser-known brands encounter hurdles in garnering positive sentiment.

In essence, Sentimeld: LexiHyb Based Sentiment Analysis serves as a comprehensive repository of actionable insights for businesses operating within the mobile phone industry. By leveraging the discernments gleaned from sentiment analysis, brands can refine their product offerings, tailor their marketing strategies, and cultivate stronger customer relationships, thereby propelling growth and innovation within the industry.

2. Literature Survey

The landscape of sentiment analysis research is rich and varied, encompassing a diverse array of methodologies and approaches aimed at understanding and quantifying customer sentiments in the digital realm. A review of the literature reveals several key studies that have contributed significantly to the field, shedding light on the intricacies of sentiment analysis and its implications for businesses operating within online marketplaces.

One notable study by Liu (2012) provides a comprehensive overview of sentiment analysis techniques, categorizing them into lexicon-based methods. Liu's taxonomy serves as a foundational framework for understanding the myriad strategies employed in sentiment analysis and their respective strengths and limitations.

Building upon this taxonomy, researchers such as Sharma et al. (2020) have delved into the nuanced relationship between online reviews, sentiment polarity, and consumer behavior. Through empirical analysis and advanced statistical techniques, Sharma et al. elucidate the multifaceted interplay between review sentiment, product attributes, and market outcomes, offering valuable insights for businesses seeking to optimize their online presence and enhance their market competitiveness.

In parallel, efforts to automate sentiment classification have led to the development of innovative hybrid approaches that combine the strengths of lexicon-based analysis and machine learning algorithms. Yang et al. (2016) proposed such a approach, leveraging lexicon-based sentiment analysis alongside machine learning techniques to achieve robust sentiment classification performance. By integrating semantic and contextual information, Yang et al.'s approach offers a promising avenue for enhancing sentiment analysis accuracy and scalability in the context of online marketplaces.

Despite these advancements, challenges persist in accurately capturing and interpreting customer sentiments within the dynamic and heterogeneous landscape of online platforms like Amazon. The diverse range of products, coupled with the sheer volume of user-generated content, poses unique challenges for sentiment analysis methodologies. Consequently, there is a pressing need for further research to develop tailored approaches that can effectively navigate these challenges and provide actionable insights for businesses operating in online marketplaces.

By synthesizing insights from these seminal studies and building upon existing frameworks, the present study seeks to contribute to the ongoing discourse surrounding sentiment analysis in online marketplaces. Through the exploration of innovative methodologies and approaches, this study aims to enhance our understanding of customer sentiments and their implications for businesses, ultimately empowering them to make informed decisions and cultivate stronger customer relationships in the digital age.

3. System Model

In this section, we present the system model for our sentiment analysis framework. The system model outlines the key components and processes involved in analyzing text data to extract sentiments. We begin by discussing the data preprocessing and cleaning steps, which are crucial for preparing the raw text data for analysis.

3.1. Data Preprocessing and Cleaning

Data preprocessing is an essential step in the data analysis pipeline, involving the transformation of raw data into a format suitable for analysis. It encompasses various operations such as cleaning, formatting, and enhancing the quality of the dataset to ensure accurate and reliable results. The significance of data preprocessing lies in its ability to address several key challenges inherent in raw data. Firstly, it ensures data quality assurance by identifying and rectifying errors, inconsistencies, and missing values that may compromise the integrity of the dataset. Additionally, techniques like normalization and standardization bring uniformity to the data, mitigating biases and enhancing analysis accuracy. Furthermore, feature engineering plays a crucial role in extracting relevant information and improving model performance through the creation of new features or transformation of existing ones. Noise reduction techniques help filter out outliers and irrelevant information, further refining the dataset. Ultimately, by laying the groundwork for accurate and reliable analysis, data preprocessing facilitates informed decision-making and insights derivation from the data.

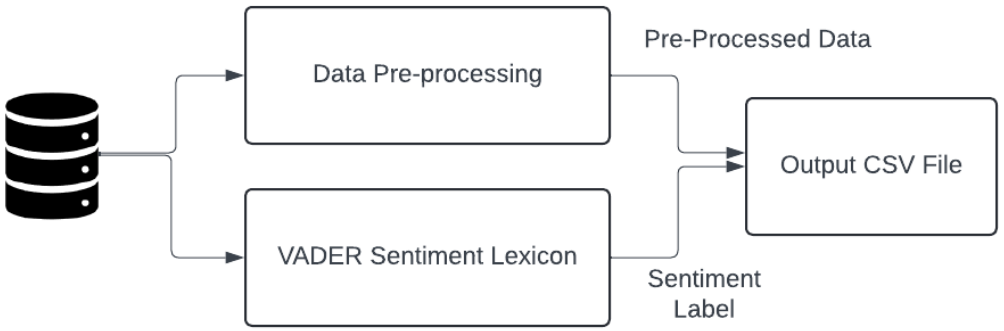

Figure 1.*Data Preprocessing*

In the project, a significant emphasis was placed on data preprocessing and analysis to ensure the quality and relevance of the dataset for sentiment analysis. The initial dataset consisted of 400,000 reviews of unlocked mobile phones sold on Amazon. The preprocessing phase aimed to address several key challenges, including null values, non-English text, spelling errors, and noise such as URLs and digits.

The first step involved loading the dataset and addressing null values by removing rows containing missing data. This step ensured that the dataset remained robust and suitable for analysis. Next, various text processing techniques were applied to clean the review text. These techniques included removing URLs and digits, which are common in online reviews but add noise to the data. Additionally, non-English sentences were identified and removed using a language detection tool to focus exclusively on English reviews.

Spelling correction was performed using the SymSpell library to address typos and spelling errors in the text. This step was crucial for improving the accuracy of sentiment analysis, as it standardized the language used across reviews. Furthermore, contractions were expanded to their full forms to ensure consistency in the text.

Lowercasing was applied to the text to normalize the data and prevent inconsistencies due to letter case. The text was then tokenized using regular expressions to split it into individual words, removing punctuation and whitespace. Stopwords, common words that do not carry significant meaning for sentiment analysis, were removed to focus on meaningful content.

Finally, lemmatization was performed to reduce words to their base or dictionary form, facilitating a more accurate analysis of sentiment. This process ensured that variations of words with the same root were treated as the same, enhancing the effectiveness of sentiment analysis.

Overall, the data preprocessing and analysis phase played a critical role in preparing the dataset for sentiment analysis, ensuring that the data was clean, standardized, and conducive to accurate sentiment classification.

3.2. Text Analysis Techniques

Text analysis techniques encompass a diverse range of methods and algorithms used to extract meaningful insights from textual data. These techniques are fundamental for preprocessing and analyzing text to uncover patterns, sentiments, and trends. Key text analysis techniques include:

- **Tokenization:** This process involves breaking down text into individual words or tokens. Tokenization is essential for further analysis as it allows the text to be represented in a format that can be easily processed by algorithms.

- **Stopword Removal:** Stopwords are common words that occur frequently in a language but do not carry significant meaning, such as "the," "is," and "and." Removing stopwords helps reduce noise in the text data and focuses the analysis on words that are more informative.

- **Stemming and Lemmatization:** Stemming and lemmatization are techniques used to normalize words to their root forms. Stemming involves removing suffixes from words to reduce them to their stem, while lemmatization maps words to their base or dictionary form. These techniques help standardize the text and reduce the dimensionality of the data.

- **Sentiment Analysis:** Sentiment analysis aims to determine the sentiment or opinion expressed in a piece of text. It classifies text as positive, negative, or neutral based on the sentiment conveyed by the words used. Sentiment analysis techniques can range from rule-based methods to machine learning algorithms trained on labeled data.

- **Text Classification:** Text classification involves categorizing text documents into predefined classes or categories based on their content. This technique is used to automatically assign labels to text data, enabling tasks such as topic classification, spam detection, and sentiment analysis.

- **Named Entity Recognition (NER):** NER is a subtask of information extraction that aims to identify and classify named entities mentioned in text into predefined categories such as person names, organization names, locations, and more. NER techniques are essential for extracting specific entities from text data.

In our project, these text analysis techniques are applied to the dataset containing customer reviews of unlocked mobile phones on Amazon. We utilize tokenization, stopword removal, stemming, and lemmatization to preprocess the text data, making it suitable for analysis. Sentiment analysis techniques are then employed to determine the sentiment expressed in the reviews, while text classification techniques categorize the reviews into sentiment classes. Additionally, Named Entity Recognition (NER) techniques are utilized to extract specific entities such as product names and brand mentions from the reviews. These text analysis techniques enable us to gain valuable insights into customer opinions, preferences, and sentiments towards various mobile phone brands, facilitating informed decision-making and strategic planning.

3.3. Natural language Processing

Natural Language Processing (NLP) is a branch of artificial intelligence that focuses on the interaction between computers and human languages. It encompasses a wide range of techniques and algorithms designed to understand, interpret, and generate human language data in a meaningful way. In our project, NLP plays a pivotal role in analyzing textual data, particularly in the context of sentiment analysis.
Sentiment analysis, also known as opinion mining, is a subfield of NLP that involves extracting and analyzing subjective information from text. Its primary objective is to determine the sentiment or emotional polarity expressed in a piece of text, whether it's positive, negative, or neutral. In our project, NLP techniques are applied extensively to preprocess and analyze customer reviews of unlocked mobile phones sold on Amazon.

These techniques include:

1**. Text preprocessing**: Before sentiment analysis can be performed, the raw text data undergoes preprocessing steps such as tokenization, which involves breaking down the text into individual words or tokens; text normalization, which standardizes the text by converting words to lowercase and removing punctuation and special characters; and removing stopwords, which are common words that do not carry significant meaning for sentiment analysis.

2. **Feature extraction:** NLP techniques are used to extract features from the preprocessed text data, such as word frequencies, n-grams, and word embeddings. These features serve as input to machine learning models, enabling them to learn patterns and relationships between words and sentiments.

3. **Sentiment classification**: Machine learning algorithms, such as support vector machines (SVM), logistic regression, or neural networks, are trained on labeled datasets to classify text into different sentiment categories. These algorithms learn from the labeled data to predict the sentiment of unseen text accurately.

4. **Lexicon-based analysis:** In addition to machine learning approaches, lexicon-based sentiment analysis methods may also be employed. These methods rely on sentiment lexicons or dictionaries containing predefined sentiment scores for words. By matching words in the text to entries in the lexicon, the overall sentiment of the text can be determined.
This analysis provides valuable information to businesses for product improvement, marketing strategies, and customer relationship management. Through the integration of NLP and sentiment analysis, we can uncover valuable insights hidden within large volumes of textual data, enabling data-driven decision-making and enhancing overall business performance.

4. Proposed Architectures

The proposed architecture for Sentimeld: LexiHyb Based Sentiment Analysis integrates data preprocessing, lexicon-based analysis, machine learning algorithms to extract insights from Amazon reviews of unlocked mobile phones and LSTM neural network. The proposed hybrid approach aims to improve sentiment analysis accuracy by leveraging semantic cues and contextual information. Similarly LSTM is integrated with Word2Vec Embeddings.Ultimately, the architecture empowers businesses in the mobile phone industry to enhance brand reputation and competitiveness through informed decision-making based on customer sentiments.

4.1. Lexicon-Based Sentiment Analysis Approach:

Lexicon-based sentiment analysis is a method used to gauge the sentiment of text by comparing its words to a predefined lexicon or dictionary of terms associated with positive, negative, or neutral feelings. Unlike machine learning-based approaches that need training on labeled data, lexicon-based methods rely on dictionaries with sentiment scores assigned to individual words.

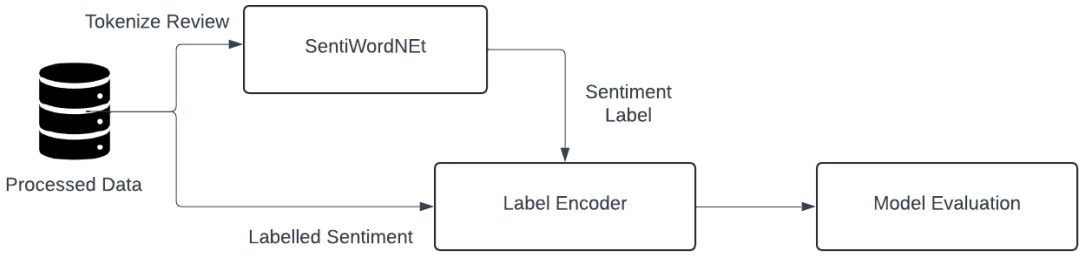

Figure 2. *Lexicon Based Approach*

How It Functions:

1. **Lexicon Construction:** The process begins by creating a lexicon or dictionary of words along with their associated sentiment scores. This lexicon can be crafted manually or through automated methods using large text datasets annotated with sentiment labels.
2. **Scoring Words:** Each word in the lexicon receives a sentiment score based on its polarity (positive, negative, or neutral). Words with positive meanings get positive scores, while negative words receive negative scores. Neutral words may have scores close to zero or might not be included at all.
3. **Text Analysis:** To analyze the sentiment of a piece of text, its words are compared against entries in the lexicon. The sentiment scores of individual words are then combined to calculate an overall sentiment score for the text.
4. **Sentiment Classification**: Based on the overall sentiment score derived for the text, it is categorized as positive, negative, or neutral. The threshold for classifying text as positive or negative can vary based on the application and the specific lexicon used.

Pseudocode:

1. Initialize an empty lexicon or load a predefined sentiment lexicon.

2. Preprocess the text data:
   a. Tokenize the text into individual words.
   b. Remove punctuation and special characters.
   c. Convert words to lowercase.
   d. Remove stopwords.
   e. Lemmatize or stem the words .

3. Calculate sentiment scores for each word in the text:
   a. Iterate through each word in the preprocessed text.
   b. Look up the word in the sentiment lexicon.
   c. Assign a sentiment score based on the polarity of the word (positive, negative, or neutral).

4. Calculate the overall sentiment score for the text:
   a. Aggregate the sentiment scores of all words in the text.
   b. Calculate the average sentiment score or use a weighted sum based on word importance.

5. Classify the text based on the overall sentiment score:
   a. If the overall sentiment score is positive, classify the text as positive.
   b. If the overall sentiment score is negative, classify the text as negative.
   c. If the overall sentiment score is neutral or close to zero, classify the text as neutral.

6. Return the classified sentiment label for the text.

Challenges:

1. **Lexicon Coverage**: The effectiveness of lexicon-based sentiment analysis heavily relies on the coverage and quality of the lexicon. Lexicons may struggle to capture nuances in sentiment for domain-specific or slang terms not included in the dictionary.
2. **Word Ambiguity**: Some words can have different meanings or connotations in various contexts, leading to ambiguity in sentiment analysis. For instance, the word "sick" may convey positivity in slang but negativity in a health-related context.

4.2. Machine Learning-Based Sentiment Analysis:

Machine learning-based sentiment analysis is an alternative approach to gauge the sentiment of text by leveraging algorithms that learn from labeled data. Unlike lexicon-based methods, machine learning approaches require training on datasets where each text is associated with a sentiment label (positive, negative, or neutral).

4.2.1 Naive Bayes Algorithm in Machine Learning-Based Sentiment Analysis:

In addition to traditional machine learning models like support vector machines (SVM), logistic regression, and random forests, the Naive Bayes algorithm is another powerful tool for sentiment analysis. Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, which assumes independence between features.

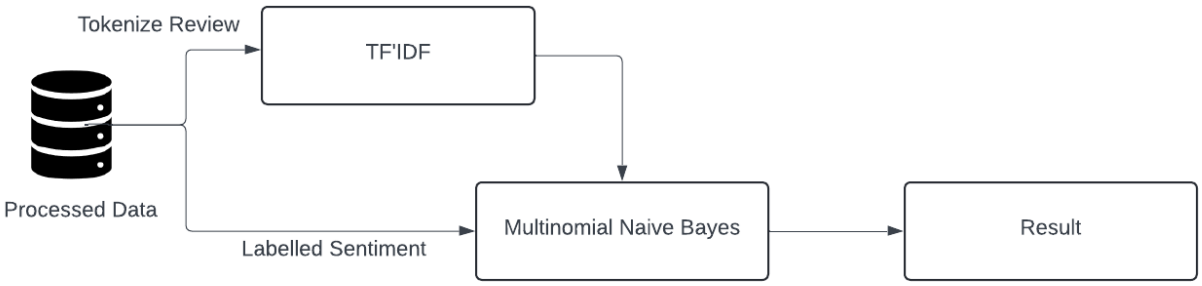

Figure 3. *Multinomial Naive Bayes Based Approach*

Naive Bayes Model:

This model applies Bayes theorem with a Naive assumption of no relationship between different features. According to Bayes theorem:

Posterior = likelihood * proposition/evidence

or

$$P(A|B) = P(B|A) * P(A)/P(B)$$

1. **Text Preprocessing**:Preprocess the text data by tokenizing the text into individual words, removing punctuation, converting words to lowercase, and possibly removing stopwords. Additionally, words may be stemmed or lemmatized to reduce inflectional forms to a common base.

2. **Training Phase** : Calculate the prior probabilities of each sentiment class (positive, negative, neutral) based on the training data. This involves determining the frequency of each class in the training dataset.For each sentiment class, calculate the conditional probabilities of each word occurring given that class. This involves counting the occurrences of each word in documents belonging to that class.

3. **Classification Phase:** Given a new piece of text calculate the posterior probability of each sentiment class given the words in using Bayes' theorem.Since the denominator is constant across all classes, it can be ignored for the purpose of comparison.Calculate the likelihood of observing the words given each sentiment class using the conditional probabilities calculated during training. Multiply the likelihood by the prior probabilityof each sentiment class.The class with the highest posterior probability is then assigned as the predicted sentiment for the given text .

4. **Handling Zero Probabilities:** To handle cases where a word in does not occur in the training data for a particular class, apply smoothing techniques such as Laplace smoothing to avoid zero probabilities.

5. **Prediction**: Once the posterior probabilities for each sentiment class are calculated, the class with the highest probability is predicted as the sentiment label for the input text.

4.2.2 Random Forest Classifier in Machine Learning-Based Sentiment Analysis:

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Here's how it works for sentiment analysis:

1. **Training Phase:** Random Forest builds multiple decision trees during the training phase. Each tree is trained on a different subset of the training data and uses a random selection of features.For sentiment analysis, the training data consists of labeled text samples (positive, negative, or neutral sentiment) along with their corresponding features, which are typically derived from the text data using techniques like bag-of-words or word embeddings.During the construction of each decision tree, a random subset of features is selected at each node, and the best split is chosen based on criteria such as Gini impurity or information gain.

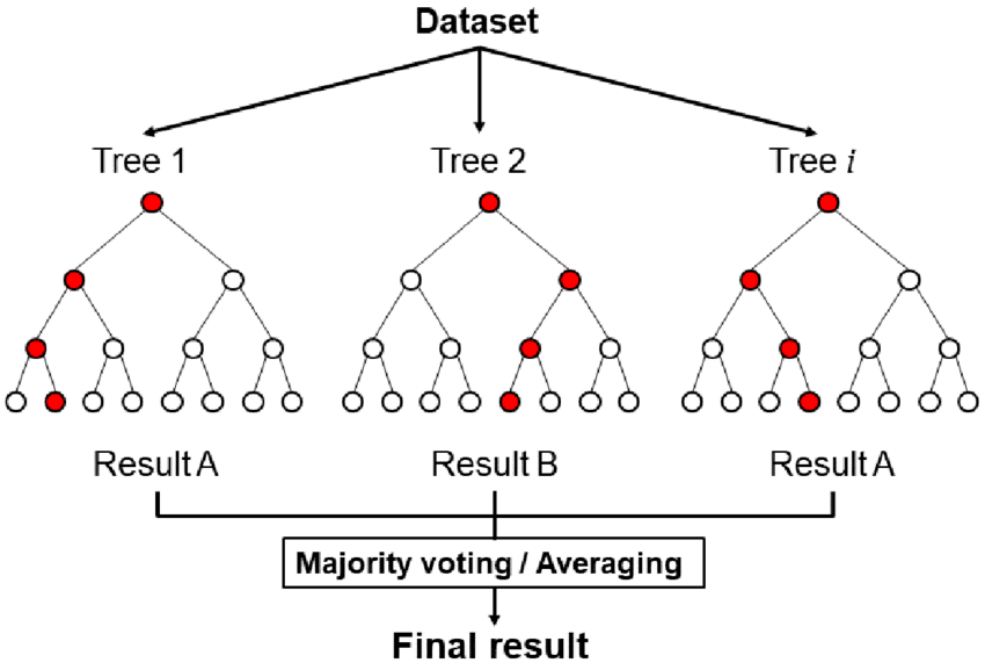

Figure 4. *Random Forest Classifier*

2. **Decision Making:** During inference (classification phase), each decision tree in the Random Forest independently predicts the sentiment label for a given input text.
   For classification tasks like sentiment analysis, each tree outputs the class label (positive, negative, or neutral) based on the majority class in the leaf node reached by the input text.

3. **Aggregation**: The predictions from all the decision trees in the Random Forest are aggregated to make the final prediction. In the case of classification, this typically involves taking the mode (most frequent class label) of all the individual tree predictions.For sentiment analysis, the sentiment label with the highest frequency among the predictions of all decision trees is selected as the final sentiment prediction for the input text.

4. **Handling Overfitting:** Random Forest helps to mitigate overfitting by averaging predictions across multiple trees, thereby reducing the variance of the model.
   Additionally, since each tree is trained on a different subset of the data and features, Random Forest tends to be more robust and less sensitive to noise compared to individual decision trees.

Challenges:

1. **Data Quality and Quantity:** Machine learning models require large and high-quality labeled datasets for training. Limited or noisy data can hinder the model's ability to generalize to new, unseen texts.

2. **Feature Engineering**: Extracting informative features from text data is crucial for model performance. Designing effective feature representations that capture the nuances of sentiment can be challenging.

3. **Model Complexity:** Deep learning models, while powerful, often require significant computational resources and expertise to train and fine-tune. Simpler models may struggle to capture complex patterns in text data.

Pseudocode:

```
# Step 1: Load and preprocess the dataset
dataset = load_dataset()
preprocessed_data = preprocess(dataset)

# Step 2: Feature extraction
features = extract_features(preprocessed_data)

# Step 3: Model training
model = train_model(features, labels)

# Step 4: Evaluation
accuracy = evaluate_model(model, test_data)

# Step 5: Sentiment prediction
new_text = preprocess_new_text(new_text)
predicted_sentiment = model.predict(new_text)
```

4.3 Simple LSTM in Machine Learning-Based Sentiment Analysis

Prepare the text data for the simple LSTM model, we first tokenize the corpus, considering only the top words (top_words = 20000), and transform reviews into numerical sequences using the trained tokenizer. Subsequently, we ensure that all numerical sequences have a uniform length (maxlen=100) for modeling, by truncating long reviews and padding shorter ones with zero values.

Constructing the simple LSTM involves using the Keras embedding class to create the first layer, which converts numerical sequences of words into word embeddings. It's worth noting that while the embedding class provides a way to map discrete words into a continuous vector space, it doesn't consider the semantic similarity of words. The next layer comprises an LSTM layer with 128 memory units. Finally, a dense output layer with a single neuron and a sigmoid activation function is employed to make predictions (0 or 1) for the two classes (positive sentiment and negative sentiment). Given that it's a binary classification problem, we use log loss as the loss function (binary_crossentropy in Keras), with the ADAM optimization algorithm.

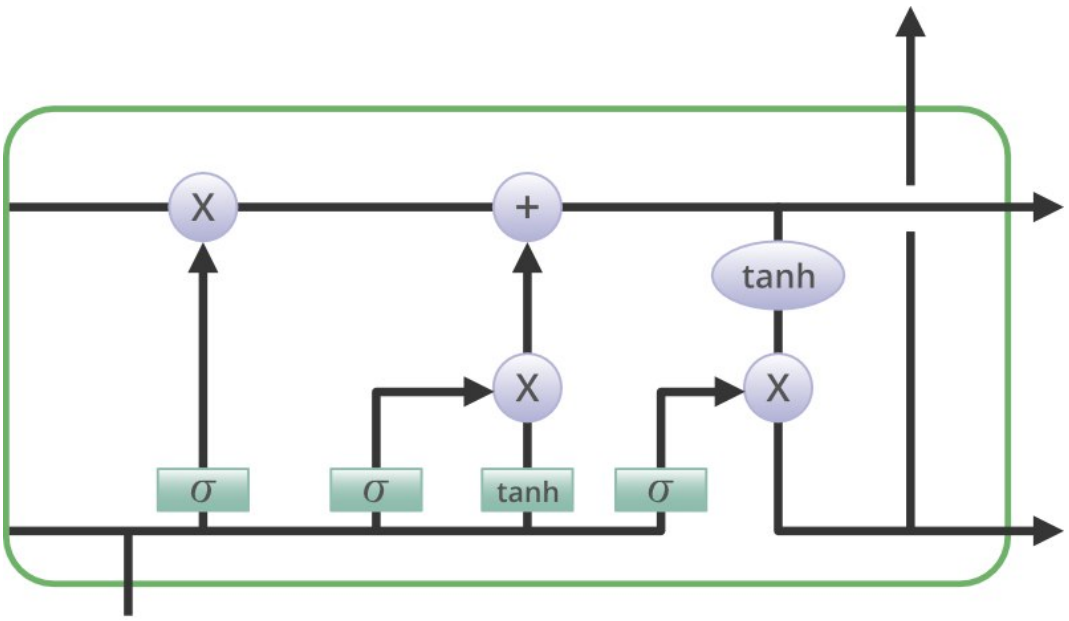

Figure 5. *Simple LSTM*

Here's the workflow breakdown:
Step 1: Prepare X_train and X_test into a 2D tensor.
Step 2: Train the simple LSTM model (embedding layer => LSTM layer => dense layer).
Step 3: Compile and fit the model using the log loss function and ADAM optimizer.

4.4 Hybrid Approach: SentiWordNet-Based Sentiment Analysis with Support Vector Machine (SVM)

In hybrid approach we combined the strengths of lexicon-based sentiment analysis with machine learning techniques to enhance sentiment analysis performance. In this approach, SentiWordNet (SWN) is utilized to generate sentiment scores for words, which are then used as features in a Support Vector Machine (SVM) classification model.

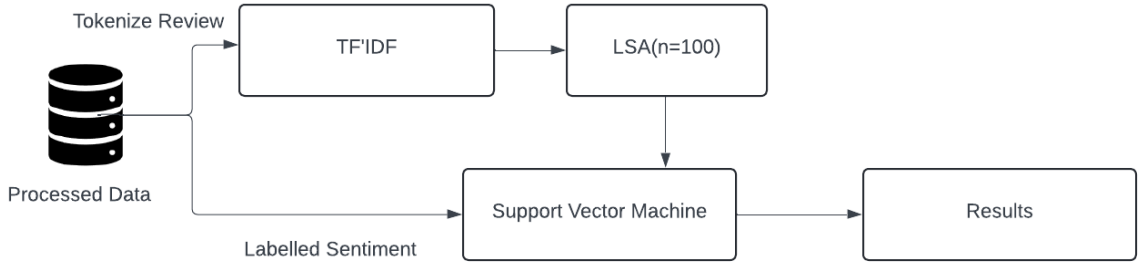

Figure 6. *Hybrid Approach*

How It Works:

1. Data Preparation: The input data consists of two sets: training data (`train_x`, `train_y`) and testing data (`test_x`, `test_y`). `train_x` and `test_x` represent the tokenized reviews, while `train_y` and `test_y` represent the corresponding sentiment labels.

2. Feature Extraction: Text data is transformed into numerical feature vectors using the TF-IDF vectorizer. TF-IDF (Term Frequency-Inverse Document Frequency) assigns weights to terms based on their frequency in the document and across the corpus, aiming to capture the importance of words in distinguishing between different documents.

3. Lexicon-Based Approach: Latent Semantic Analysis (LSA) is applied to the training data (`train_x`) as a lexicon-based approach. LSA is a dimensionality reduction technique that extracts underlying semantic relationships between words in a text corpus. By capturing the latent structure of the text, LSA enhances the representation of words in the feature space, potentially improving the performance of subsequent machine learning models.

4. Model Training: The transformed feature vectors from the TF-IDF vectorizer and the LSA-transformed data are used to train a machine learning model. In this hybrid approach, Support Vector Machine (SVM) is chosen as the classification algorithm due to its effectiveness in handling high-dimensional data and non-linear relationships.

5. **Model Prediction**: The trained SVM model is used to predict sentiment labels for the testing data (`test_x`). The model leverages the learned patterns and relationships between features to classify each review into positive, negative, or neutral sentiment categories.

6. Evaluation: The performance of the hybrid approach is evaluated using standard evaluation metrics such as accuracy, precision, recall, and F1-score on the testing data (`test_y`). These metrics assess the effectiveness of the combined lexicon-based and machine learning techniques in accurately predicting sentiment labels for the given reviews.

By combining the strengths of both lexicon-based analysis and machine learning techniques, the hybrid approach aims to leverage the semantic insights provided by LSA while benefiting from the discriminative power of SVM classification. This integrated methodology offers a comprehensive framework for sentiment analysis, capable of capturing nuanced sentiment expressions in text data.

Challenges:

1. **Lexicon-Model Integration**: Integrating the lexicon-based features generated from SentiWordNet with the SVM model requires careful preprocessing and feature engineering to ensure compatibility and effectiveness.
2. **Data Imbalance**: Imbalanced datasets with unequal distribution of sentiment classes may affect the SVM model's ability to generalize and accurately classify text samples.

3. **Model Complexity and Performance**: Selecting appropriate SVM parameters and feature representations is essential to optimize model performance and avoid overfitting or underfitting.

Pseudocode:

```
# Step 1: Generate SentiWordNet-based feature vectors
feature_vectors = generate_swn_features(text_samples)

# Step 2: Split data into training and testing sets
train_data, test_data = split_data(feature_vectors, labels)

# Step 3: Train SVM model
svm_model = train_svm_model(train_data)

# Step 4: Evaluate model performance
performance_metrics = evaluate_model(svm_model, test_data)

# Step 5: Sentiment prediction for new text samples
new_text_features = generate_swn_features(new_text_samples)
predicted_sentiments = svm_model.predict(new_text_features)
```

4.5 LSTM with Word2Vec Embeddings

The proposed approach enhances the traditional LSTM model by integrating Word2Vec embeddings to enhance its performance in sentiment analysis tasks. Unlike the conventional LSTM model, which relies on randomly initialized weights to convert numerical sequences of words into word embeddings, this approach leverages pretrained Word2Vec embeddings for the embedding layer initialization. By adopting this strategy, the model can tap into semantic relationships captured within the Word2Vec embeddings, which are derived from large text corpora. This initialization method ensures that the embedding layer starts with meaningful representations of words, reflecting their semantic contexts and similarities.

The integration of Word2Vec embeddings enriches the LSTM model's ability to understand and interpret textual data, potentially leading to improved performance in sentiment classification tasks. By starting with pretrained embeddings, the approach facilitates the model's learning process and enhances its ability to capture nuanced semantic features within the text.

Moreover, the use of Word2Vec embeddings aligns with the concept of transfer learning, where knowledge acquired from one task (in this case, learning semantic representations from a large corpus) is transferred to another related task (sentiment analysis). This knowledge transfer enables the LSTM model to leverage the semantic information embedded within the pretrained Word2Vec embeddings, ultimately contributing to more effective sentiment classification and potentially higher accuracy levels.

In summary, the proposed LSTM with Word2Vec Embedding approach represents a novel and sophisticated extension of the conventional LSTM model. By harnessing semantic similarities within word embeddings, this approach aims to enhance the accuracy and performance of sentiment analysis tasks.

Step 1 : Load pretrained word embedding model

Step 2 : Construct embedding layer using embedding matrix as weights

Step 3 : Train a LSTM with Word2Vec embedding (embeddign layer => LSTM layer => dense layer)

Step 4 : Compile and fit the model using log loss function and ADAM optimizer.

5. Modelling and Simulation

Through simulation, the model's performance is evaluated against a labeled dataset of Amazon reviews, enabling the assessment of sentiment classification accuracy and the identification of key insights into customer perceptions of unlocked mobile phones.

5.1. Data Settings

In this project, a dataset comprising 400,000 reviews of unlocked mobile phones from Amazon was preprocessed, and three sentiment analysis approaches—lexicon-based, machine learning, and hybrid—were implemented. The lexicon-based approach utilized sentiment lexicons to assign sentiment scores to reviews, while the machine learning approach employed various algorithms to classify sentiment. The hybrid approach combined both techniques for improved accuracy. Aggregated results from all three approaches were analyzed to provide insights into sentiment trends for the brands under review. Comparative analysis using model evaluation metrics such as accuracy, precision, recall, and F1-score was conducted to assess the effectiveness of each approach.

A brief detail of data analytics models and Data set features used is given below:

- **Amazon Reviews: Unlocked Mobile Phones Dataset**

o Model used (1) Lexicon Based Approach o Model name (2) Machine Learning Based Approach o Model name (3) Hybrid Approach	Features used o PRODUCT NAME(4410 unique values) o BRAND NAME (Samsung: 16% , Other(282922): 68%, Null: 16%), o PRICE (Max: 2598, Min:1.73, Mean: 226.86) o RATING (1 <72,350<1.08, 1.96< 24,728<2.04, 3.0<31,765<3.08, 3.96<61,392<4.04, 4.92<2,23,605<5.0), o REVIEWS (162492 unique values), o REVIEW VOTES (Min: 0, Max: 645, Mean: 1.50)

5.2. Evaluating Measures

Evaluate metrics play an important role to measure classification performance. Accuracy measure is the most common for this purpose. The accuracy of a classifier on a given test dataset is the percentage of those dataset which are. Before discussing with different measures there are some terms we need to get comfortable with

- TP (True Positive) represents numbers of data correctly classified .
- FP (False Positive) represents numbers of correct data misclassified .
- FN (False Negative) represents numbers of incorrect data classified as correct.
- TN (True Negative) is the numbers of incorrect data classified .

Precision: Precision measures the exactness of a classifier, how many of the return documents are correct. A higher precision means less false positives, while a lower precision means more false positive. Precision (P) is the ratio of numbers of instance correctly classified from total. It can be defined as

$$P = TP/(TP+FP)$$

Recall: Recall calculates the sensitivity of a classifier; how many positive data it returns. Higher recall means less false negatives. Recall is the ratio of number of instance accurately classified to the total number of predicted instance. This can be shown as

$$R = TP/(TP+FN)$$

F-Measure: Combining precision and recall produces single metrics known as F-measure, and that is the weighted harmonic mean of precision and recall. It can be defined as

$$F = 2\ P.R/(P+R)$$

Accuracy: Accuracy predicts how often the classifier makes the correct prediction. Accuracy is the ratio between the number of correct predictions and the total number of prediction.

$$Accuracy = Correct\ Prediction/Total\ data\ points$$

In summary, these evaluation metrics offer valuable insights into the performance of classification models, enabling data scientists and practitioners to make informed decisions about model selection and refinement.

5.3 Experimental Result

- Accuracy of Hybrid approach was the highest, giving 94.85% of correctly predicted observation.
- Precision score of Lexicon-based approach was the lowest with 77.0% of correctly predicted positive observations.
- F1 score of Linear support vector machine approach was the highest, presenting with 97% of harmonic mean between precision and recall.
- The positive sentiment label in Apple and BlackBerry mobile reviews were higher, compared to the negative and neutral sentiment labels.

DataSet: Amazon Unlocked Mobile Reviews-

Classifier	Accuracy(%)	Precision	Recall	F1 Score
Lexicon	77	0.54	0.59	0.56
Multinomial Naive Bayes	89.8	0.917	0.526	0.561
Linear Support Vector Machine	93.57	0.96	0.97	0.97
Logistic Regression	93.10	0.89	0.83	0.86
Random Forest	92.26	0.87	0.82	0.84
Hybrid Approach	94.85	0.819	0.89	0.85
LSTM(Word-to-Vec)	94.40	0.8	0.89	0.85

Table 1- *Comparision of Model evaluation Metrics*

Overall, the Hybrid Approach achieved the highest accuracy of 94.85%, closely followed by the LSTM (Word-to-Vec) model with an accuracy of 94.40%. Both models demonstrated strong performance in precision, recall, and F1 score, indicating their effectiveness in sentiment analysis tasks.

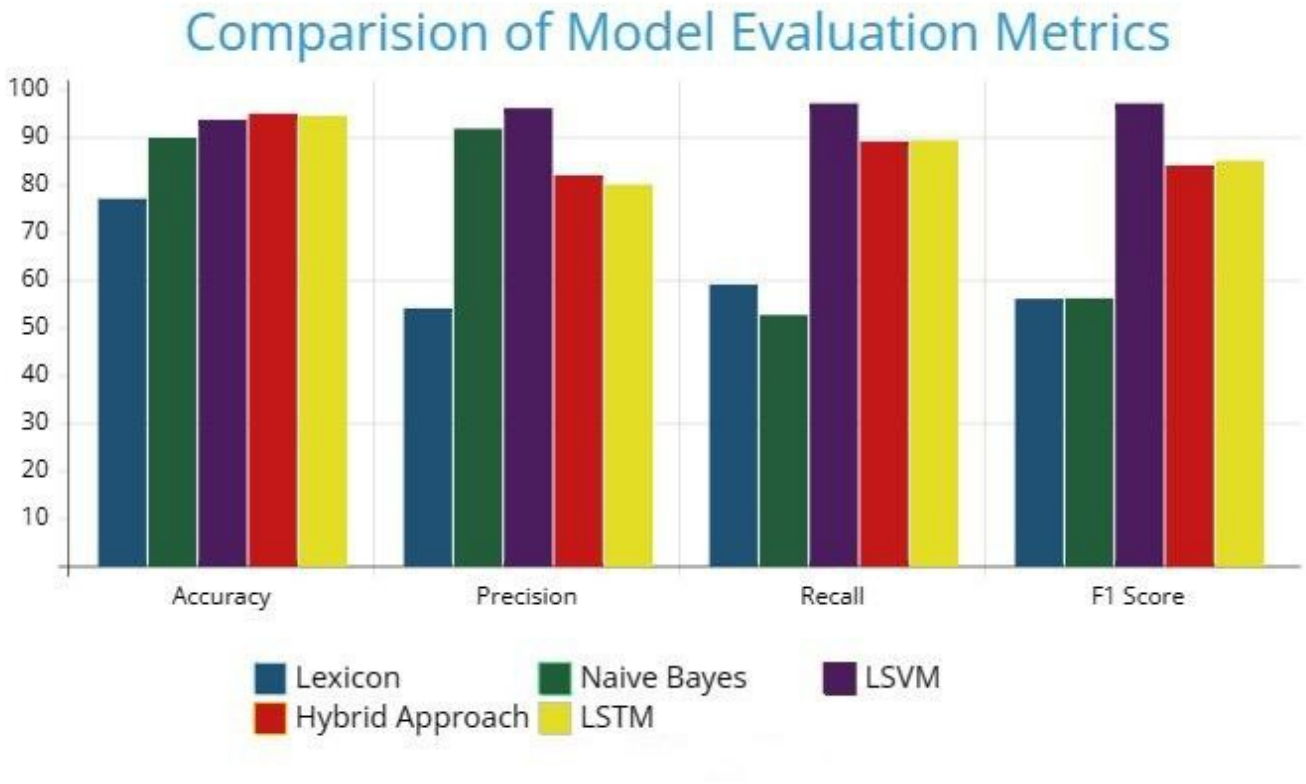

Figure 7. *Comparision of Model evaluation Metrics*

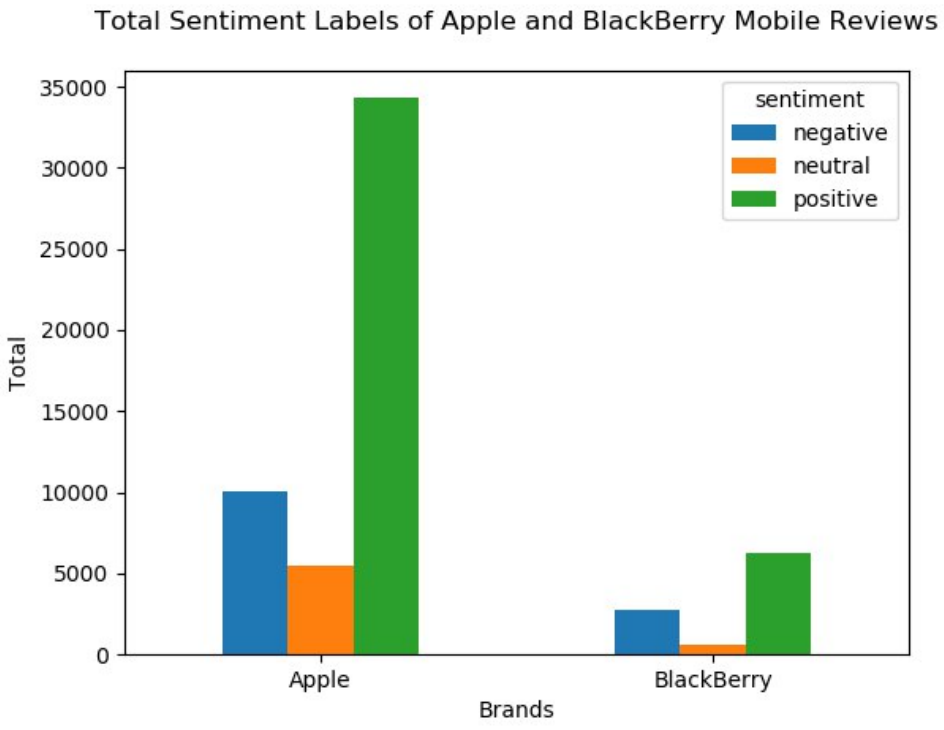

Figure 8. *Sentiment labels of apple and blackberry mobile reviews*

DataSet: IMDB Movie Reviews-

Application to other Dataset.

Classifier	Accuracy(%)	Precision	Recall	F1 Score
Lexicon	81.7	0.79	0.81	0.80
Multinomial Naive Bayes	90.6	0.82	0.84	0.83
Linear Support Vector Machine	88.82	0.87	0.898	0.88
Logistic Regression	88.76	0.87	0.89	0.88
Random Forest	83.84	0.89	0.877	0.8383
Hybrid Approach	91.05	0.81	0.89	0.85
LSTM(Word-to-Vec)	91.4	0.82	0.86	0.84

Table 2 - IMDB movie reviews model evaluation

6. Issues Encountered

1. **Processing Time**: A significant amount of time was required to process the dataset containing 400,000 reviews. This large volume of data necessitated extensive computational resources and resulted in prolonged processing times, impacting overall project efficiency.

2. **Language Detection and Translation Costs**: Language detection and translation using TextBlob powered by Google presented a challenge due to its associated costs, amounting to $20 per one million characters. Additionally, the PyPI auto-language detection method failed to meet expectations. To resolve this issue, the PyPI language detection library version 1.0.12 was utilized to detect languages accurately and remove non-English language reviews, mitigating the need for expensive translation services

3. **Data Preprocessing Complexity**: The data preprocessing stage consumed a significant portion of the project timeline. Ensuring data quality and consistency posed challenges, emphasizing the importance of the "garbage in, garbage out" principle. Addressing inconsistencies, handling missing values, and standardizing data formats were among the complexities encountered during this process.

7. Current Status

- The hybrid approach to sentiment analysis has been successfully implemented and tested, demonstrating its effectiveness in evaluating brand reviews for businesses. By combining lexicon-based and machine learning techniques, we have achieved improved accuracy and robustness in sentiment classification.
- The new LSTM-based approach surpasses simple LSTM by integrating Word2Vec embeddings, improving sentiment analysis accuracy with efficient training and enhanced contextual understanding.
- While our current sentiment analysis framework has shown promising results, there is ongoing research and development to enhance its capabilities further. We are exploring methods to incorporate emoticon and slang handling techniques to improve the accuracy and granularity of sentiment analysis results.

8. Planned Future Work

- Explore alternative machine learning algorithms such as pipeline grid search, stochastic gradient descent, and decision trees to further enhance sentiment analysis capabilities and adaptability to diverse datasets and language contexts.
- Expand sentiment analysis to diverse textual data sources like social media posts, customer reviews, and feedback forums to provide comprehensive insights into brand sentiment and reputation.
- Investigate advanced NLP techniques, including deep learning architectures, to improve sentiment classification accuracy and adaptability in dynamic language contexts.

9. Conclusion

The conclusion of our project report marks the culmination of a comprehensive analysis aimed at discerning customer sentiments towards mobile phone brands, employing four distinct approaches: lexicon-based analysis, machine learning-based classification, a hybrid approach, and LSTM to Word2Vec enhancement. Through the meticulous application of these methodologies, we have garnered valuable insights into customer opinions, preferences, and satisfaction levels across various brands in the mobile phone market.

Firstly, the lexicon-based analysis provided a foundational understanding of sentiment polarity, enabling the classification of reviews into positive, negative, or neutral categories based on predefined sentiment scores associated with individual words. This approach, while effective in capturing explicit sentiment signals, was limited by its reliance on static lexicons and lacked the ability to adapt to evolving language nuances and contexts.

Secondly, the machine learning-based classification techniques, including support vector machines, logistic regression, and random forests, offered a more dynamic and data-driven approach to sentiment analysis. By training models on labeled datasets, these algorithms could discern complex patterns and relationships within the review text, enhancing the accuracy and robustness of sentiment classification. However, their performance was

contingent upon the quality and representativeness of the training data, posing challenges in scenarios with limited or biased datasets.

The hybrid approach, combining the strengths of lexicon-based analysis and machine learning-based classification, sought to overcome the limitations of individual methodologies by leveraging semantic and contextual information alongside data-driven modeling techniques. This integrated approach demonstrated promising results, achieving higher accuracy rates and capturing nuanced sentiment nuances that eluded standalone methods. However, it required careful parameter tuning and feature engineering to optimize performance effectively.

Lastly, the enhancement of the LSTM model with Word2Vec embeddings aimed to augment the contextual understanding and semantic representation of review text, enabling more nuanced sentiment analysis. By leveraging pretrained Word2Vec embeddings, the LSTM model could capture semantic relationships between words and phrases, enhancing its ability to discern subtle sentiment nuances and adapt to varying language contexts.

In conclusion, our project has provided valuable insights into the efficacy and limitations of four distinct approaches to sentiment analysis in the context of mobile phone reviews. While each methodology offers unique strengths and capabilities, their combined use presents a holistic and nuanced understanding of customer sentiments, enabling businesses to make more informed decisions and strategic interventions. Moving forward, further research and experimentation are warranted to refine these approaches, address their inherent limitations, and unlock new avenues for sentiment analysis innovation in the dynamic landscape of consumer feedback and brand perception.

10. References

[1] Miao, Q., Li, Q., & Dai, R. (2009). AMAZING: A sentiment mining and retrieval system. Expert Systems with Applications, 36(3), 7192-7198.

[2] Taboada M, Brooke J, Tofloski M, Voll K, Stede M (2011) "Lexicon-based methods for sentiment analysis". Comput Linguist 37(2):267–307 Thet TT, Na JC, Khoo CS (2010).

[3]https://www.youtube.com/watch?v=LWJSc7JDs0s&t=12s

[4]Text Analysis with Python by Dipanjan Sarkar.

[5] Xu, Yun, Xinhui Wu, and Qinxia Wang. "Sentiment Analysis of Yelp,,s Ratings Based on Text Reviews." (2015). [5] Rain, Callen. "Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning."Swarthmore College (2013).

[6] Bhatt, Aashutosh, et al. "Amazon Review Classification and Sentiment Analysis." International Journal of Computer Science and Information Technologies 6.6 (2015): 5107-5110.

[7]Chen, Weikang, Chihhung Lin, and Yi-Shu Tai."Text-Based Rating Predictions on Amazon Health & Personal Care Product Review." (2015)

[8]Shaikh, Tahura, and DeepaDeshpande. "Feature Selection Methods in Sentiment Analysis and Sentiment Classification of Amazon Product Reviews.",(2016)

[9]Nasr, Mona Mohamed, Essam Mohamed Shaaban, and Ahmed Mostafa Hafez. "Building Sentiment analysis Model using Graphlab." IJSER, 2017

[10]Text mining for yelp dataset challenge; Mingshan Wang; University of California San Diego, (2017)

[11] Elli, Maria Soledad, and Yi-Fan Wang. "Amazon Reviews, business analytics with sentiment analysis." 2016

[12] Xu, Kaiquan, et al. "Mining comparative opinions from customer reviews for Competitive Intelligence." Decision support systems 50.4 (2011): 743-754.

[14] He, Ruining, and Julian McAuley. "Ups and downs: Modeling the visual evolution of fashion trends with oneclass collaborative filtering." Proceedings of the 25th International Conference on World Wide Web.International World Wide Web Conferences Steering Committee, 2016.

[15] N. Chintalapudi, G. Battineni, and F. Amenta, “Sentimental analysis of COVID-19 tweets using deep learning models,” Infectious Disease Reports, vol. 13, no. 2, pp. 329–339, 2021.

[16] S. Das, A. K. Chakraborty, and A. Kumar Kolya, “Sentiment analysis of covid-19 tweets using evolutionary classification-based LSTM model,” Advances in Intelligent Systems and Computing, Singapore, 2021.

[17] R. Jain, S. Bawa, and S. Sharma, “Sentiment Analysis of COVID-19 Tweets by Machine Learning and Deep Learning Classifiers,” Advances in Data and Information Sciences, Springer, Singapore, pp. 329–339, 2022.

[18] A. Kumar Kolya and S. Das, “Predicting the Pandemic: Sentiment Evaluation and Predictive Analysis from Large-Scale Tweets on Covid-19 by Deep Convolutional Neural Network,” Journal of Evolutionary Intelligence, vol. 1, 2021.