

Improving Sentiment Analysis of Shopee Reviews with Informal Language and Slang

Ahmad Hariz Imran bin Ahmad Azrir, Naveen Palanichamy*, Su-Cheng Haw, Kok-Why Ng

Faculty of Computing and Informatics, Multimedia University (MMU), Cyberjaya, Malaysia.

p.naveen@mmu.edu.my (Corresponding author)

Abstract. Sentiment analysis (SA) is essential for businesses seeking to understand customer preferences and opinions. It aims to identify and interpret the emotions present in the text by analyzing its linguistic patterns. Along with the difficulty of comprehending formal textual content, it is also essential to consider the informal and online social media languages, which combine English with regional slang to express peoples' genuine feelings. The presence of slang and informal language in customer reviews can compromise the accuracy of SA. This study attempts to improve the accuracy of SA when informal and slang are present, specifically in the context of Shopee customer reviews. The approach employs feature extraction techniques such as N-grams, Term Frequency-Inverse Document Frequency (TF-IDF), and Bag of Words (BoW). Machine learning algorithms such as Support Vector Machine (SVM), Naive Bayes (NB), Gated Recurrent Unit (GRU), and Long Short-Term Memory (LSTM) are then used to analyze the sentiments of Shopee customer reviews. The results show that the SVM classifier using the 2-gram TF-IDF features outperformed other algorithms. In addition, the study found that incorporating a slang dictionary into SA models improved the classification of informal language and slang, resulting in higher accuracy in sentiment analysis.

Keywords: Sentiment analysis, Shopee, feature extraction, machine learning, slang dictionary, Deep learning

1. Introduction

Shopee is a well-known online marketplace in Southeastern Asia that has gained significant traction in recent times. The platform enables merchants to establish virtual stores and offer their products to millions of customers across the region. Shopee has made a remarkable impact on the e-commerce landscape in Southeast Asia, providing an effortless shopping experience with a diverse range of products at competitive prices. Its success can be attributed to its intuitive user interface, convenient payment options, and an efficient customer support system that ensures a hassle-free shopping journey for buyers. Consequently, Shopee has emerged as a dominant player in the e-commerce industry, continuously expanding its user base throughout the region.

The analysis of sentiment has become an invaluable tool for comprehending customer opinions and feedback within the e-commerce sector. The proliferation of online shopping platforms like Shopee has intensified the need for businesses to grasp the sentiment conveyed in customer reviews and feedback. This is particularly significant in the present digital era, where customers have the ability to express their opinions online and share their experiences with a vast audience. Employing machine learning models, it is feasible to automatically categorize the sentiment of a review based on its linguistic components. The research community has displayed increasing interest in the development of effective techniques for sentiment analysis. However, most of these methods have predominantly focused on English content. Furthermore, existing research primarily concentrates on the formal written English language, whereas customer reviews often include informal language and slang expressions, posing challenges to accurate sentiment classification. Therefore, this study endeavours to explore the complexities of addressing informal language in sentiment analysis through an experiment conducted on customer reviews obtained from Shopee. By addressing the obstacles presented by colloquial language, we strive to provide more precise insights into customer sentiment on online platforms.

The structure of this research paper unfolds as follows: Section 2 presents a collection of relevant studies in the field of sentiment analysis. In Section 3, we provide an overview of the methods employed in this experiment. Moving on to Section 4, we elaborate on the experiment's details and present our findings. Finally, Section 5 concludes this paper, summarizing the key takeaways.

2. Literature Review

In this literature review, we will analyze some of the existing methods and techniques used in sentiment analysis.

2.1. Handling Informal Language

The challenges posed by informal language within sentiment analysis can be met with a variety of preprocessing techniques, extending beyond the immediate application of features and machine learning models. Abbreviation expansion, correction of misspellings, and language standardization emerge as effective strategies to consider. Furthermore, the incorporation of machine learning models trained on datasets containing a broad spectrum of slang words and informal expressions becomes pivotal for achieving accurate sentiment classification, especially within reviews featuring such expressions.

One promising work, explored by Sunitha (2022), involves the utility of a slang dictionary. The integration of such a dictionary aid in identifying slang words and can lead to their substitution with standard counterparts or the attribution of sentiment scores based on associated sentiments within the dictionary. This underscores the importance of leveraging external linguistic resources to enhance the precision of sentiment analysis.

Building upon these fundamental approaches, additional insights into handling slang and informal language can be drawn from related works. Research that explores context-aware sentiment analysis offers methodologies for understanding the nuanced meanings of slang words within specific contexts. Techniques such as word embeddings, as demonstrated by Smith and Eisenstein (2021), prove valuable in capturing the semantic subtleties of slang.

2.2. Feature extraction

Text preprocessing techniques, including stemming and stopword elimination, are widely employed to cleanse and prepare textual data before feeding them into sentiment analysis models. However, the impact of these techniques on the accuracy of sentiment analysis remains a subject of ongoing discussion. In a recent research article (Pradana, 2019), the authors focused on assessing the effects of preprocessing methods on sentiment analysis accuracy in the context of Indonesian language texts. They conducted experiments using various preprocessing conditions, both with and without stemming and stopword removal, and applied a Support Vector Machine classifier with TF-IDF weighting. Interestingly, their findings indicated that the inclusion of stemming and stopword removal had only a minimal influence on the accuracy of sentiment analysis for Indonesian text documents.

In another study (Iswanto & Poerwoto, 2018), researchers examined the impact of stemming and stopword elimination on sentiment analysis accuracy for Indonesian text documents, specifically focusing on sentiment analysis of Twitter data. Their automated sentiment analysis approach achieved an impressive accuracy and recall rate of up to 85.50%. Surprisingly, the study discovered that these preprocessing techniques had a negligible impact on sentiment analysis accuracy. Additionally, in a separate investigation (Tyagi, et al., 2019), Tweepy was employed to extract Twitter data, and sentiment categorization (positive, negative, or neutral) was performed using a K-Nearest Neighbor algorithm with N-gram modeling. Jalani et al. (2022) conducted a study where they analyzed user sentiment towards three clothing brands by examining brand mentions on Twitter and applying the TextBlob algorithm to classify tweets into three categories based on polarity.

Web scraping is a commonly utilized technique for automating data extraction from websites. In an academic paper Mussalimun, E. H. (2021), the researchers adopted web scraping techniques, leveraging the BeautifulSoup4 Python library, to extract restaurant reviews from the Tripadvisor platform. The extracted data encompassed various attributes such as the restaurant name, reviewer's name, comment reviews, and ratings. However, for their specific project, only the restaurant name and customer reviews were utilized. To preprocess the extracted data, the researchers employed the Sastrawi Python library, which is a functional library designed for the Indonesian language. The Sastrawi library facilitated stemming, a process that reduces words to their base form by removing inflections and variations. Additionally, NLTK functions were applied to tokenize the data and remove stopwords, which are commonly occurring words in a language that carry little contextual meaning. The resulting preprocessed data was subsequently employed for sentiment analysis to determine the sentiment expressed in the customer reviews.

Table 1 summarizes the feature extraction methods used in various papers. It demonstrates the wide range of techniques employed by researchers, such as TF-IDF, N-Gram, Bag-of-Words (BoW). Among these methods, TF-IDF is the most commonly used feature extraction technique across multiple studies followed up by bag of words. This indicates its popularity and effectiveness in capturing important textual features for sentiment analysis. Additionally, the table highlights the flexibility and diversity in selecting feature extraction techniques for sentiment analysis, allowing researchers to adapt their approaches to specific study requirements.

Table 1. Summary of the feature extraction methods

Study	TF-IDF	Relief	MVO	N-Gram	BoW	FastText	TextBlob	Vader
Pradana (2019)	✓							
Hassonah, et al. (2020)	✓	✓	✓					
Tyagi, et al. (2019)				✓				
Harish, et al. (2019)	✓				✓			

Saad & Yang (2019)	✓							
Kristiyanti, et al. (2023)						✓		
Yin, et al. (2021)	✓							
Iswanto & Poerwoto, (2018)				✓				
Baid, et al. (2017)	✓				✓			
Matsumoto, et al. (2018)	✓				✓			
Aljedaani, et al. (2022)	✓				✓		✓	
Sunitha, D. P. (2022)	✓					✓		
Widjaja, J. A. (2018)								✓
Yang, S. E. (2019).							✓	✓
Venkateswarlu Bonta, N. K. (2019)	✓				✓			
Jalani et al. (2022)							✓	

2.3. Supervised Machine learning

In a recent publication (Fitri, Andreswari, & Hasibuan, 2019), researchers employed the Naive Bayes algorithm to classify the sentiment polarity of Twitter users within the Indonesian context. Surprisingly, most comments regarding the Anti-LGBT campaign were found to be neutral, achieving an impressive accuracy rate of 86.43% using the Naive Bayes algorithm. Similarly, another study (Yin, et al., 2021) introduced and evaluated two algorithms, namely Naive Bayes and Random Forest, on datasets comprising Malay Twitter comments enriched with internet slang and abbreviated forms. Intriguingly, the Random Forest algorithm outperformed Naive Bayes as a classifier, achieving an accuracy of 81.25%.

The findings of these studies showcased diverse outcomes, with some reporting high accuracy rates in sentiment analysis, while others indicated relatively lower rates. For instance, research (Pradana, 2019) demonstrated that the inclusion of stemming and stopword removal had a marginal impact on sentiment analysis accuracy for Indonesian text documents. Conversely, study (Fitri, Andreswari, & Hasibuan, 2019) highlighted that the Naive Bayes algorithm outperformed Decision Trees (DT) and Random Forest (RF) models in sentiment analysis accuracy, revealing a predominance of neutral comments regarding the Anti-LGBT campaign. Additionally, investigation (Saad & Yang, 2019) identified the DT algorithm as the most accurate in detecting ordinal regression, while research (Kristiyanti, et al., 2023) revealed that the integration of Fasttext Embedding enhanced sentiment analysis accuracy when compared to conventional techniques. Moreover, study (Yin, et al., 2021) highlighted the superiority of the Random Forest classifier over Naive Bayes, emphasizing the efficacy of supervised machine learning algorithms in sentiment analysis accuracy improvement. In a distinct approach, research (Baid, et al., 2017) introduced an innovative method utilizing automatically annotated category labels based on emojis, which outperformed traditional word feature-based techniques. Likewise, investigation (Iswanto & Poerwoto, 2018) determined that Naive Bayes classifiers employing unigram feature models achieved optimal performance without the need for stopword elimination or stemming, with an impressive accuracy and recall rate of up to 85.50%. Finally, based on the insights of research Sairamvinay Vijayraghavan, D. B. (2020), both K-Nearest Neighbors (K-NN) and Naive Bayes classification exhibited comparable accuracy, precision, and recall rates, with

Naive Bayes slightly surpassing K-NN by a 2% margin in accuracy and precision.

In addition, researchers have also explored hybrid models that combine various feature extraction and selection techniques to enhance sentiment analysis accuracy. In a recent publication (Hassonah, et al., 2020), a hybrid approach integrating Support Vector Machines with feature selection techniques ReliefF and MVO achieved significant improvement in accuracy (up to 96.85%) compared to existing methods when applied to over 6900 tweets. Similarly, another study (Harish, et al., 2019) focused on sentiment analysis of IMDb movie reviews and employed a hybrid feature extraction method combining TF and TF-IDF with a lexicon corpus, resulting in superior results compared to classifiers such as SVM, Naive Bayes, KNN, and Maximum Entropy.

2.4. Deep learning Models

In the study outlined in paper (Matsumoto, et al., 2018), the primary focus is on the classification of emojis used in tweets, utilizing deep learning methods. The researchers conducted a comparative analysis of several neural network models, including Feed Forward Neural Network, CNN, BiLSTM RNN, and BiGRU. Remarkably, the study found that models built upon the BiLSTM architecture achieved the highest accuracy. Furthermore, the authors employed evaluation metrics such as precision, recall, and F1 score, in addition to a confusion matrix, to assess the models' performance. This investigation sheds light on the potential of leveraging deep learning techniques to classify non-textual elements, such as emojis, within social media data.

The subsequent paper (Aljedaani, et al., 2022) introduces a hybrid sentiment analysis approach that combines deep learning models with lexicon-based techniques to enhance sentiment accuracy. The researchers evaluated the classification accuracy of various machine learning models, including LR, RF, SVC, DT, GBC, CNN, LSTM, GRU, and LSTM-GRU. Additionally, they compared the effects of TextBlob, Afinn, and VADER as sentiment lexicons. To extract features, TF-IDF and BoW were utilized as feature extraction methods. Notably, the results demonstrated that the LSTM-GRU model outperformed all other models, achieving an impressive accuracy of 0.97 and an F1 score of 0.96. This study underscores the significance of integrating diverse techniques to improve sentiment analysis accuracy. Paper Sunitha, D. P. (2022). proposes a sentiment analysis model that specifically focuses on tweets related to the coronavirus. The researchers collected 3100 tweets from European and Indian individuals within a specific timeframe and employed TF-IDF, GloVe, pre-trained Word2Vec, and quick text embedding techniques to preprocess the data and extract features. Subsequently, they employed an ensemble classifier consisting of a GRU and a CapsNet to categorize users' emotions as fear, joy, sadness, and rage. The experimental findings showcased the proposed model's ability to classify the emotions of Indian and European individuals with prediction accuracies of 97.28% and 95.20%, respectively. This study emphasizes the potential of utilizing deep learning techniques to categorize emotions within social media data related to ongoing events, such as the COVID-19 pandemic. Drawing insights from the results presented in paper Venkateswarlu Bonta, N. K. (2019), the conclusion is reached that deep learning algorithms outperform other machine learning algorithms in predicting sentiment within reviews. Notably, the highest accuracy scores were achieved by deep learning algorithms, specifically ANN, LSTM, and GRU, when using TF-IDF features in the context of depression conditions. The ANN model exhibited the best performance, with an accuracy score of 90.1993%, closely followed by LSTM at 90.6085%, and GRU at 90.0162%. Conversely, SVM, Logistic Regression, and RF algorithms achieved comparatively lower accuracy scores, with SVM scoring the highest at 77.6737%, followed by LogReg at 74.2811%, and RF at 61.1632%. The superior performance of deep learning algorithms, particularly neural networks, is attributed to their ability to capture essential features crucial for sentiment classification within reviews. Furthermore, the study concludes that employing Countvectorizer encoding alongside deep learning algorithms yields superior performance compared to other models. It is also worth noting that both LSTM and GRU models exhibit similar performance across various conditions, indicating that recurrent neural networks possess comparable

capabilities.

Table 2 presents a summary of the machine learning algorithms utilized in various research papers. It highlights the prevalence of the usage of algorithms such as SVM and NB, which have demonstrated their effectiveness across different domains. Furthermore, deep learning models such as LSTM and GRU, are also used for sentiment analysis tasks. For our project, we will focus on employing SVM and NB as our supervised machine learning algorithms, while exploring LSTM and GRU for our deep learning models.

3. Methodology

Figure 1. The study aims to analyze customer reviews of Shopee, a popular online shopping platform. Data is extracted from Shopee's app on Google Play using the Google-Play-Scraper API, and pre-processing techniques are applied to ensure that the dataset is clean and ready for analysis. A lexicon-based method is used to label each review with a polarizing value, indicating the sentiment of the review. The dataset is then divided into training and testing sets, and feature extraction methods such as BoW, N-gram, and TF-IDF are applied to process the data. Machine learning models are used to predict the sentiment of new customer reviews, including both deep learning and traditional machine learning algorithms. The performance of these models is evaluated using metrics such as accuracy, precision, recall, and F1-score. By analyzing the sentiment of Shopee's customer reviews, valuable insights can be gained into areas for improvement in the platform's user experience.

Table 2. Summary of Machine learning algorithms used

Paper	SVM	NB	KNN	MaxE	DT	RF	LSTM	GRU	SVC	GBC
Pradana (2019)	✓									
Hassonah, et al. (2020)	✓									
Tyagi, et al. (2019)	✓	✓	✓							
Harish, et al. (2019)	✓	✓		✓						
Fitri, et al. (2019)		✓			✓	✓				
Saad & Yang, 2019)					✓	✓				
Kristiyanti, et al. (2023)	✓	✓								
Yin, et al. (2021)						✓				
Iswanto & Poerwoto, 2018)	✓	✓		✓						
Baid, et al. (2017)		✓								
Matsumoto, et al. (2018)							✓	✓		
Aljedaani, et al., 2022)					✓	✓	✓	✓	✓	✓
Widjaja, J. A. (2018).	✓	✓		✓						
Sunitha, D. P. (2022).								✓		
Venkateswarlu Bonta, N. K. (2019)	✓				✓	✓			✓	✓

Sairamvinay Vijayraghavan, D. B. (2020)		✓	✓						
Mussalimun, E. H. (2021)			✓						

3.1. Data Collection

For this task, we will be implementing the Google Play Scraper API to extract reviews for a specific app from the Google Play Store. To begin, install the "google-play-scraper". Next, our target is to scrape reviews for the "Shopee" app, we use the package name "com.shopee". Additionally, the parameters are set to only extract English reviews so that we do not encounter any mixed languages in our dataset. Then, the reviews are gathered in batches of 200 per loop. Next, we sort the reviews to prioritize the most relevant ones. Utilizing the capabilities of the Google Play Scraper API, we retrieve reviews in the form of dictionaries containing valuable details such as 'reviewId', 'userName', 'userImage', 'content', 'score', and 'date'. To preserve this essential data, we store it in a CSV file, which serves as input for training and evaluating our machine learning models.

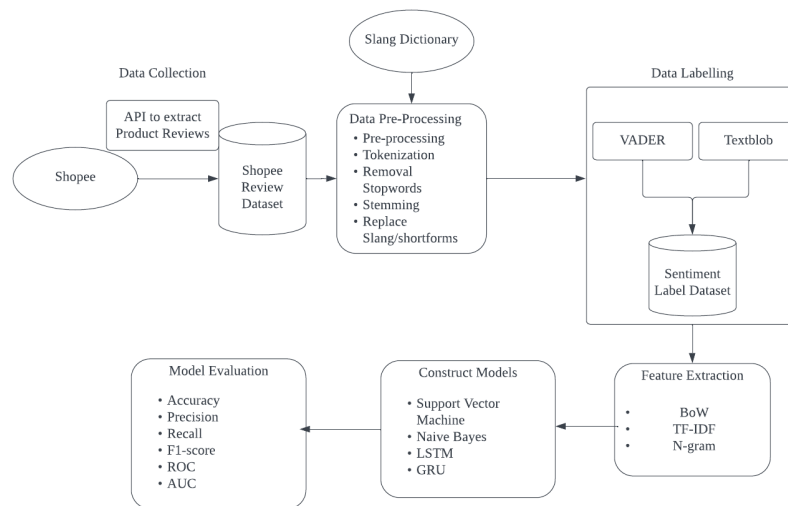


Fig 1: Proposed architecture

3.1.1. Dataset

The characteristics of the dataset:

- Username: The name or pseudonym of the user who wrote the review. It provides information about the reviewer's identity or chosen display name.
- Content: The textual content of the review itself, where users express their opinions, experiences, or feedback regarding the Shopee app.
- Score: The rating or score given by the reviewer, indicating their overall satisfaction with the Shopee app. It typically ranges from 1 to 5, with 5 being the highest satisfaction level.
- Thumbs Up Count: The number of thumbs-up or positive reactions received by the review from other users. It reflects the popularity or agreement with the expressed sentiments.
- At: The user mentioned or tagged in the review, if applicable. This provides information about interactions or discussions involving specific users.
- Reply Content: The content of the reply or response to the review, if any. It represents the interaction between the user and the app developer or customer support team.
- Replied At: The date and time when the reply to the review was made. It indicates the timeframe for addressing user concerns or providing assistance.

- Sort Order: The sorting order or position of the review in relation to other reviews, typically based on criteria such as relevance, recency, or popularity.
- App ID: The unique identifier for the Shopee app. It allows for tracking and organizing reviews specifically related to the Shopee app.

The dataset captures a collection of reviews that users have written about the Shopee app. These reviews serve as valuable resources for sentiment analysis tasks, allowing for the development and evaluation of models that can analyze the sentiment or opinion expressed by users towards the Shopee app. By considering the textual content, ratings, and other characteristics of the reviews, sentiment analysis models can learn to identify positive, negative, or neutral sentiments associated with the app, which can provide insights into user satisfaction and potential areas for improvement.

For this reserach, we have created a new column called target and reviews with a score higher than 4 will be given 0 as their target and 1 if it is less. The target column will be used to predict whether the review is positive or negative later in the machine learning section of the work. Then, the dataset is stored in a CSV (Comma-Separated Values) file format, where each row represents a review and the columns correspond to the characteristics mentioned above. This file format facilitates easy storage, retrieval, and analysis of the review data, enabling researchers and practitioners to process and manipulate the dataset efficiently.

3.2. Data Pre-processing

The initial step involves data collection through an API and saving it in a CSV file. After that, data pre-processing is conducted, encompassing tasks such as exploratory data analysis, handling null values, and transforming the dataset by dropping unnecessary columns and modifying data types.

To facilitate standard pre-processing techniques, several adjustments are implemented, including the removal of hashtags, mentions, numerical values, non-alphabetic symbols, case folding, URLs, emojis, and hyperlinks. Tokenization is then applied to break down the dataset into smaller tokens, such as individual words or phrases, extracted from the original dataset. The removal of stop words eliminates commonly used words that do not contribute significant information to tasks like sentiment analysis. Additionally, stemming simplifies words by removing prefixes, suffixes, and other grammatical components that are not essential for understanding the word's meaning.

Next, text pre-processing is the step where it involves a series of modifications to the text data to enhance manageability. The approach encompasses standard pre-processing techniques, tokenization, stop word removal, and stemming. Firstly, the 'BeautifulSoup' library is employed to eliminate HTML tags from the raw text. Regular expressions are then utilized to remove numbers, non-letter characters, punctuation, and multiple spaces from the text. The resulting output is a cleaned and prepared text suitable for analysis. To ensure accuracy, a slang dictionary is established to replace slang terms in the dataset with their corresponding formal translations. The slang dictionary can be found through kaggle where it has been modified with the latest slangs from the internet. Lastly, emoticons and emojis are substituted with words representing their associated sentiments, aiding in the expression of emotions and sentiments.

3.2.1. Slang Dictionary

The slang dictionary mentioned, which is sourced from Kaggle and extracted from Urban Dictionary, is a dataset specifically curated to capture a wide range of informal language, including slang words, expressions, acronyms, and their corresponding meanings. Urban Dictionary is a crowdsourced online platform that allows users to contribute definitions for various slang terms and phrases. The slang dictionary consists of more than 3000 acronyms and slang words that are mostly used online. Regarding sentiment analysis, the incorporation of a slang dictionary involves the utilization of its contents to establish associations between the slang terms and their conventional English meanings. The dataset consists of a pairing of a term or acronym with its corresponding standard or expanded form.

For instance:

- woke, "aware and socially conscious"
- ootd, "outfit of the day"
- Dope, "awesome"
- "Lit", "exciting"
- "hella", "very"

By integrating a colloquialism reference into the preprocessing workflow, sentiment analysis models become proficient in handling and accurately interpreting casual language frequently found in Shopee reviews. This inclusion considerably augments the precision of sentiment analysis outcomes, ensuring accurate interpretation and analysis of the text despite the presence of unmeaningful expressions and slang.

3.3. Data Labelling

Textblob holds a crucial position in sentiment analysis for this project as it employs a lexicon-based approach to assess the sentiment of textual data. According to research outlined in paper Widjaja, J. A. (2018), the utilization of Textblob enables the calculation of polarity scores for each review within the dataset. These scores, ranging from -1 to 1, provide an indication of the overall sentiment conveyed in the text.

Similarly, VADER is another tool that can be used for this project. It utilises a lexicon-based approach specifically designed for social media texts and it employs a pre-built sentiment lexicon that contains a wide range of words and their associated sentiment scores. These scores reflect the positivity, negativity, and neutrality of words, allowing VADER to assess the sentiment of a given text. Understanding the sentiments expressed by customers through reviews is pivotal for businesses, as it assists in making informed decisions. The absence of either one would pose challenges in comprehending and analyzing the sentiments conveyed in the text, ultimately compromising the accuracy of the sentiment analysis. In order to select which one is better, paper Yang, S. E. (2019). reports experimental results that support the superiority of Textblob over Vader as it yields a higher accuracy. Hence, in this project, Textblob stands as an indispensable tool for conducting sentiment analysis.

3.4. Feature Extraction

The Bag of Words (BoW) technique serves as a straightforward method for feature extraction from text data. It works by creating a histogram of all words present in the text, treating each word as an individual feature, and utilizing the frequency of occurrence of each word as a function within the training set. The implementation of the BoW technique involves employing the CountVectorizer function from Scikit-learn, which counts the tokens and generates a matrix consisting of a limited set of tokens.

The Term Frequency-Inverse Document Frequency (TF-IDF) method, on the other hand, represents another widely utilized feature extraction technique that assigns weights to words in a corpus, effectively converting text data into a numerical representation. The weight of each word is determined by multiplying its Term Frequency (TF) by its Inverse Document Frequency (IDF). TF represents the frequency of a term within a specific document, while IDF is calculated based on the number of documents that contain the term. In the context of this dataset, the TF-IDF approach employs the N-gram parameter to identify the most frequent combinations of words of size n within the text. After thorough research, our dataset suggests that using 1-gram and 2-gram ranges yield the most efficiency. Thus, for this project, the N-gram range used with the TF-IDF approach will be between 1 and 2 grams.

3.5. Supervised machine learning models

Each model in this study is constructed by combining various feature extraction techniques and specific parameter settings. The parameter configurations are detailed in Table 1 provided below. Support Vector Machines (SVMs) represent a powerful machine learning algorithm that is used for both

classification and regression tasks. The primary objective of SVMs is to identify the optimal hyperplane that can effectively separate data points into distinct classes, maximizing the margin between the closest points of different classes. SVMs demonstrate robustness against noise, and they exhibit excellent generalization capabilities when presented with new data. These characteristics make SVMs particularly suitable for handling imbalanced, high-dimensional, and non-linear datasets. However, it is important to note that SVMs can be computationally intensive and sensitive to the selection of hyperparameters. Nevertheless, SVMs stand as an effective and versatile algorithm for addressing classification and regression tasks. Naive Bayes is another widely employed supervised learning algorithm employed for data classification into multiple groups though its assumption of attribute independence often proves unrealistic for real-world datasets. To address this limitation, variants of the Naive Bayes algorithm have been developed, including Gaussian, Bernoulli, and Multinomial Naive Bayes. The algorithm estimates the probability of a new data point belonging to each category based on its attribute values and selects the class with the highest probability. This involves calculating the posterior probabilities for each class using Bayes' theorem, which takes into account the likelihood and prior probability of the data point and each class. By repeating this process for multiple data points, the algorithm can effectively classify large datasets.

Table 3. Hyperparameters used

Model	Hyperparameter
SVM	cvec__max_features: 300 cvec__min_df: 2, 3 cvec__max_df: 0.9, 0.95 cvec__ngram_range: (1, 1), (1, 2) svc__kernel: 'linear', 'poly', 'rbf' svc__degree: 3 svc__C: 0.1
NB	cvec__max_features: 500 cvec__min_df: 2, 3 cvec__max_df: 0.9, 0.95 cvec__ngram_range: (1, 1), (1, 2)

3.6. Deep Learning models

LSTM is a recurrent neural network architecture commonly employed for processing sequential data, including text data. They are effective in capturing long-term dependencies and retaining memory of past events to make predictions about future ones. The training process consists of LSTM gates adapting their weights to emphasize the most significant information within the sequence making the characteristic makes LSTMs well-suited for sentiment analysis tasks where the context and sequence of words play a crucial role in determining the sentiment expressed in a text. Thus, by combining LSTM-based RNN architectures, Shopee sentiment analysis benefits from more accurate and efficient analysis of customer feedback and reviews. Its counterpart, GRU is another type of recurrent neural network commonly employed for analyzing sequential data, it has a simpler structure and fewer parameters compared to LSTMs, which results in faster training times and easier optimization. The update and reset gates in GRUs enable the network to selectively retain relevant information in its memory, facilitating accurate predictions by capturing long-term dependencies in the data. They are used in sentiment analysis by treating the text data as a sequence of words or tokens and leveraging the gates to learn patterns indicative of different sentiments. So, the models are trained on labelled datasets of text data and can classify new and unseen data into the appropriate sentiment category.

4. Results & Discussion

4.1. Preprocessing Results

The data pre-processing involves various techniques such as exploratory data analysis, handling null values, standard pre-processing techniques such as removing hashtags, mentions, numerical values, non-alphabetic symbols, case folding, URLs, emojis, and hyperlinks are also applied, stop words removal is used to remove commonly used words, and stemming simplifies words to their fundamental form by removing suffixes and prefixes. A demo to show the before and after effects can be found in Table 2

4.1.1. Slang Dictionary Implementation

To incorporate a slang dictionary into the sentiment analysis pipeline, the process begins with reading the slang dictionary CSV file into the program. This initial step allows us to gain access to a collection of slang terms and their corresponding meanings, which will be utilized during the subsequent preprocessing stage. Once the dictionary is loaded, as depicted in Figure 2, it becomes an integral part of the workflow and is implemented following the preprocessing step. The significance of integrating a slang dictionary arises from its remarkable capability to enhance the accuracy of sentiment analysis. Slang terms and acronyms are frequently employed in informal language, particularly across social media platforms, and they possess the potential to profoundly influence sentiment interpretation. By establishing mappings between slang and their standard counterparts, the slang dictionary contributes to normalizing the text, ensuring that sentiment analysis algorithms correctly grasp the intended significance. As the dictionary becomes integrated, sentiment analysis models can effectively address the nuances of casual language, notably present in platforms like Shopee reviews. This proficiency is invaluable given that user-generated reviews often encompass informal expressions and slang. The dictionary empowers these models to precisely decode and evaluate such language, even in cases where it may seem unfamiliar or unconventional. The ultimate outcome of this integration is the substantial enhancement of sentiment analysis precision. The dictionary-driven substitution of slang with standard equivalents ensures the model's predictions are rooted in a thorough comprehension of the text, encompassing its intricacies of expression. Consequently, sentiment analysis attains heightened reliability and accuracy, successfully navigating text containing informal language and slang. This entire process underscores the dictionary's role in capturing the subtleties and intricacies of informal language, thereby enriching the quality and depth of insights derived from sentiment analysis.

```
{'acronym': 'expansion',
'2day': 'today',
'2m2h': 'too much too handle',
'2moro': 'tomorrow',
'2nite': 'tonight',
'4eae': 'for ever and ever',
'aaf': 'always and forever',
'aar': 'at any rate',
'aayf': 'as always your friend',
'abd': 'already been done',
```

Fig. 2: Slang Dictionary

Preprocessed Text	Slang Processed Text
The new album is absolutely lit the new smartphone s camera quality is goat The performance is mad fast	the new album is absolutely exciting the new smartphone s camera quality is Greatest of All Time the performance is extremely fast

Fig. 3: Example of slang processed text

Table 2 Sample reviews after data preprocessing

Preprocessing step	Review
Original	"I wonder why so often the app updates yet existing app problems were never resolved! They kept stacking up!."
General Preprocessing	'I wonder why so often the app updates yet existing app problems were never resolved They kept stacking up"
Remove stopwords	"i wonder why significant other shout out often the updates yet existing problems were never resolved they kept stacking up. "
Stem text	"i wonder why significant other shout out often the updates yet existing problems were never resolved they kept stacking up."

4.2. Comparison between Vader and Textblob

Figure 4 shows the accuracy difference between Vader and Textblob. For our dataset, Textblob achieved a higher accuracy and gave more accurate predictions as to vader. The dataset mainly consists of English reviews as when we were scraping the dataset, we have filtered out other languages so the dataset should only consist of english characters. Thus, textblob will be used as our main lexicon library.

	Textblob			Vader		
	precision	recall	f1-score	precision	recall	f1-score
0	0.81	0.75	0.78	0.79	0.73	0.76
1	0.71	0.77	0.74	0.69	0.76	0.73
accuracy			0.76			0.75

Fig. 4. Comparison between Textblob and Vader

4.3. Exploratory Data Analysis

Figure 5 shows the Number of Meaning words in Negative and Positive Reviews. This refers to the number of words in a review that convey meaning or content, rather than being filler words or stop words that convey the person’s sentiment, opinion, or description of the subject of the review. The mean of negative reviews is 48 and the mean of positive reviews is 25, this suggests that the negative reviews tend to have a higher average score than the positive reviews which means customers who leave negative reviews tend to rate the product or service lower on average than those who leave positive reviews. This may indicate that there are some significant issues or drawbacks with the product or service that are affecting the overall satisfaction of customers.



Fig. 5. Number of Meaning words in Negative and Positive Reviews

In Figure 6, the accuracy is a measure of how well a model is able to correctly predict the target variable. In this case, having a 78% accuracy on a 0.10 threshold suggests that the model is able to correctly predict the target variable 78% of the time when the Textblob compound score is above 0.10.

Figure 7 below shows the distribution of the compound scores calculated by Textblob for positive and negative reviews. The x-axis represents the range of compound scores (-1 to 1), and the y-axis shows the density of reviews falling within that range. The graph illustrates that the majority of positive reviews have a higher compound score, indicating a stronger positive sentiment, while the majority of negative reviews have a lower compound score, indicating a stronger negative sentiment. Additionally, the graph shows that there is more positive feedback compared to negative reviews within our dataset. There is also an overlap between positive and negative in the middle which could mean there are mixed reviews.

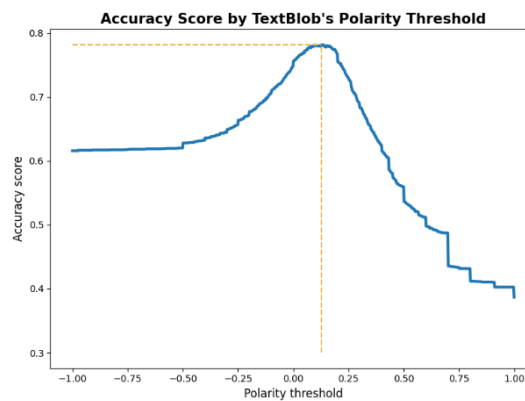


Fig. 6: Average Score by Textblob Compound Score Threshold

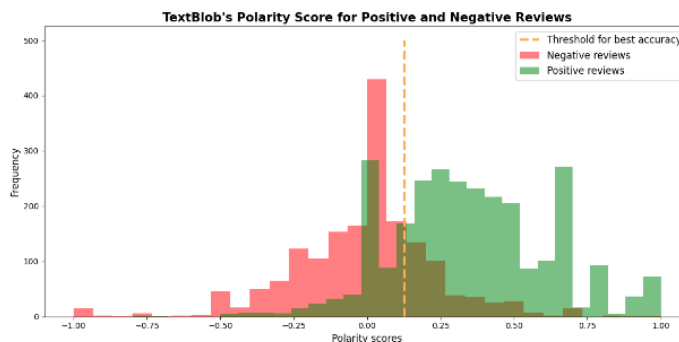


Fig. 7: Textblob Compound Score for Positive and Negative Reviews

Figure 8 shows the lists of unigrams and bigrams and reveals the most common words and phrases used in positive and negative reviews. For example, in negative reviews, words such as "time," "information," and "technology" are frequently used. In positive reviews, words such as "good," "easy," and "shopping" are frequently used. By analysing these words, we can roughly figure out what consumers feel towards the service. However, it is difficult to get the full context of the sentiment because there are lots of overlapping words that can be found in both positive and negative reviews. Thus Bigrams should be able to give a clearer picture. These phrases highlight the proper context and we can deduce that business needs to improve on customer service in order to satisfy the consumers. The following figure 9 shows the word cloud for negative and positive reviews in a different perspective as it shows which words are used more.

4.4. Supervised Machine Learning Results

The reported results, particularly when considering the integration of the slang dictionary, might seem

initially surprising due to their apparent similarity or lack of change. However, a closer examination of the "why" and "how" behind this observation can provide insights into the underlying factors.

In table 3, where the BoW technique was employed, both SVM and NB models achieved high accuracy scores, around 0.85, regardless of whether the slang dictionary was utilized. The NB models, however, demonstrated a slightly superior score of 0.87. This consistency in results might stem from the fact that BoW primarily relies on the presence or absence of words in a text and their frequency, which might not be significantly influenced by the addition of a slang dictionary. It's also possible that the dataset itself contains a substantial amount of slang terms or expressions, making the dictionary's impact less pronounced.

Moving to the second table, which employs the TF-IDF feature representation, we see a similar trend. Both SVM and NB models achieve higher accuracy scores compared to the first table. The SVM models achieve accuracy scores of 0.85 and 0.87, while NB achieves 0.88 and 0.86. Interestingly, the NB model's precision and recall metrics are also enhanced. The apparent similarity in outcomes might be attributed to the nature of TF-IDF, which considers the importance of words in a document relative to the entire corpus. While the slang dictionary can enhance the models' understanding of specific words, TF-IDF might already be assigning appropriate weights to these terms, thereby leading to consistent results.

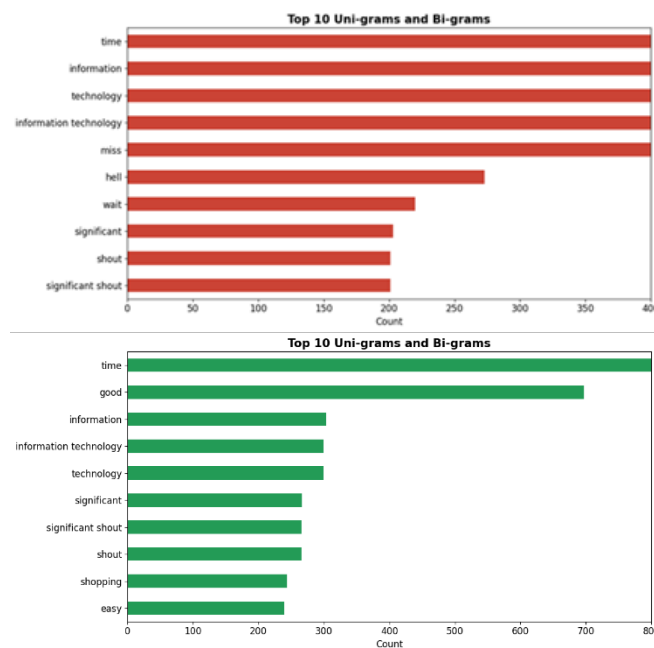


Fig. 8: Top 10 Unigrams and Bi-grams for Negative and Positive Reviews



Fig 9. Word Cloud for Negative and Positive reviews

Table 3. Performance of Machine learning with BoW

Model	Acc	Class	Precision	Recall	F1 score
SVM	0.85	0 1	0.85 0.86	0.91 0.77	0.88 0.81
SVM without slang dictionary	0.85	0 1	0.84 0.86	0.91 0.74	0.87 0.80
NB	0.87	0 1	0.90 0.83	0.89 0.84	0.89 0.83
NB without slang dictionary	0.87	0 1	0.90 0.82	0.88 0.85	0.89 0.83

Table 4 Performance of Machine learning with TF-IDF

Model	Acc	Class	Precision	Recall	F1-score
SVM	0.85	0 1	0.88 0.83	0.89 0.80	0.88 0.81
SVM without slang dictionary	0.87	0 1	0.89 0.85	0.91 0.82	0.90 0.83
NB	0.88	0 1	0.90 0.85	0.90 0.84	0.90 0.84
NB without slang dictionary	0.86	0	0.88 0.84	0.90 0.91	0.89 0.83

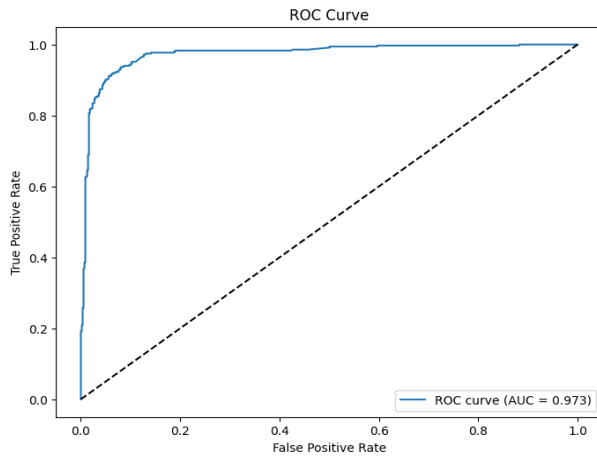


Fig. 10: ROC Curve for SVM + BoW

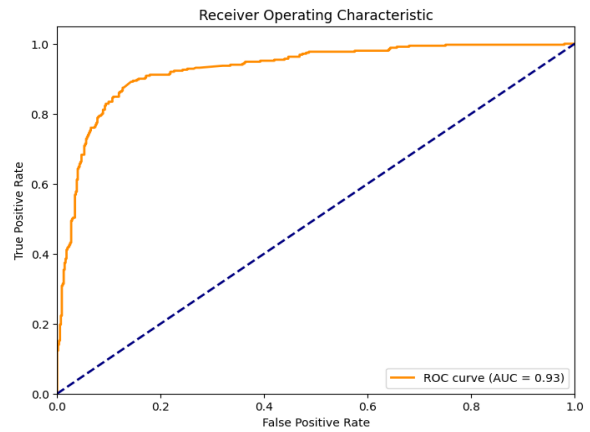


Fig.11: ROC Curve for NB + BoW

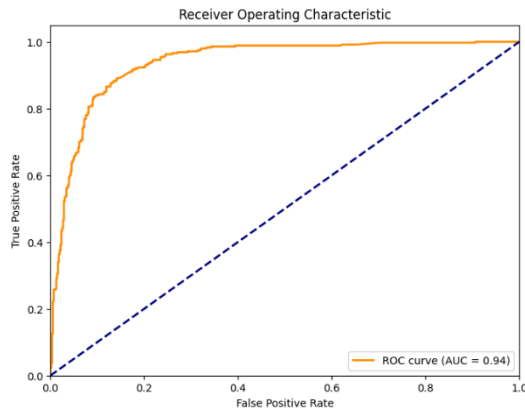


Fig.12: ROC Curve NB + TF-IDF

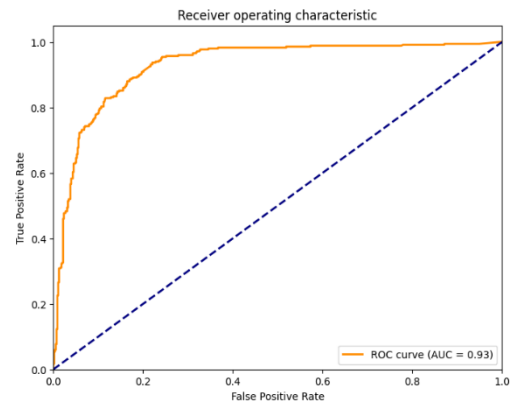


Fig.13: ROC Curve for SVM + TF-IDF

4.5. Results of Deep learning models

The models presented here are advanced deep learning architectures meticulously crafted to address the complexities of text classification tasks. These models harness the power of an embedding layer to transform each word in the input text into a vectorized representation. This vectorized data then undergoes processing via a recurrent layer. In this context, the LSTM model employs a Long Short-Term Memory layer, while the GRU model integrates a Gated Recurrent Unit layer. By leveraging these specialized layers, the models adeptly retain crucial information from previous time steps, thereby gaining a profound understanding of the contextual intricacies within the text data.

Both models employ dropout regularization techniques to mitigate overfitting risks. Additionally, they feature an output layer with a sigmoid activation function for binary classification. The primary distinction between these models lies in the number of units within the recurrent layer (32 for LSTM and 64 for GRU) and the embedding layer size (8 for LSTM and 64 for GRU). Despite these differences, the models share similar architectures and are meticulously tailored for text classification tasks.

According to Table 5, both the LSTM and GRU models achieved accuracy scores of 0.88 and 0.87, respectively. The results demonstrated a notable similarity whether or not the slang dictionary was employed. Although both models performed well, the LSTM model exhibited a slightly higher accuracy of 0.88, indicating a consistent ability to predict classes effectively. Further analysis revealed that the LSTM model also outperformed the GRU model across both positive and negative classes.

The observed trends were reinforced by Figures 14 and 15, which showcased training and validation accuracy. The LSTM model achieved higher accuracy levels, peaking at 0.93 for training and 0.89 for validation, whereas the GRU model's performance was marginally lower. This suggests that the LSTM architecture excels in sentiment analysis tasks involving sequential data. Its adeptness at capturing and preserving long-term dependencies, coupled with its gated memory cells, contributed to its superior performance. However, it's noteworthy that as the number of training epochs increased, the LSTM's validation accuracy displayed a slight decline, possibly due to insufficient training data. While the GRU model exhibited respectable performance, its simpler architecture might have constrained its ability to capture intricate data dependencies.

In conclusion, the LSTM model emerged as a more potent contender for capturing the sequential nature of text data in sentiment analysis. Nonetheless, several limitations warrant consideration when deploying these models. Contextual ambiguity can pose challenges, as the sentiment of a word or phrase often hinges on surrounding words or the sentence's overall context. The dataset analysis suggested that the use of slang is not prevalent among customers in the reviews, hinting at a higher demographic that refrains from using slang. Additionally, the models' ability to handle irony and negation is crucial, as these factors can substantially alter sentiment. Accurately discerning a speaker's underlying intent, especially in scenarios involving irony or negation, is paramount for optimal model performance in

sentiment analysis.

Table 5. Performance Evaluation

Model	Accuracy	Class	Precision	Recall	F1- score
LSTM	0.88	0 1	0.89 0.86	0.91 0.82	0.90 0.84
LSTM no slang dictionary	0.87	0 1	0.89 0.8	0.87 0.84	0.88 0.82
GRU	0.87	0 1	0.90 0.83	0.89 0.84	0.89 0.83
GRU no slang dictionary	0.87	0 1	0.87 0.81	0.89 0.79	0.88 0.81

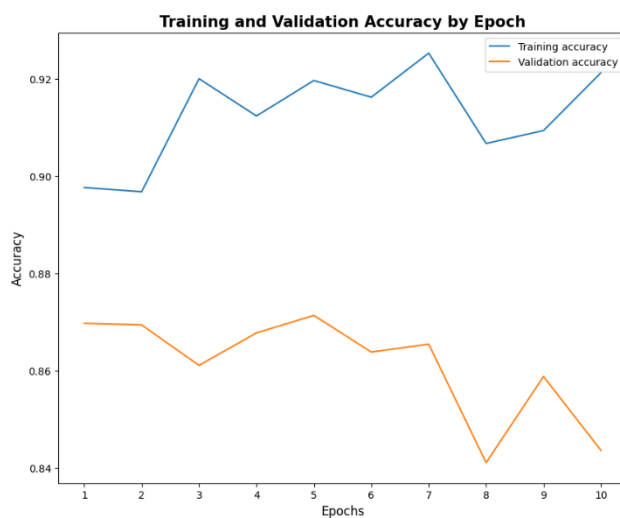


Fig 14. Training and Validation for LSTM

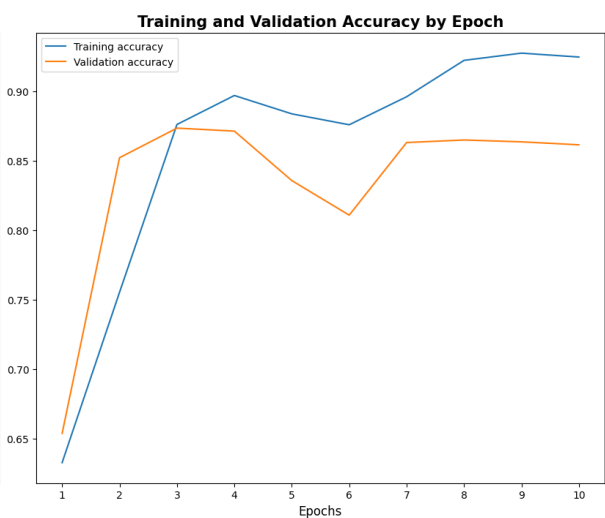


Fig 15. Training and Validation for GRU

5. Conclusion

Sentiment analysis encounters a myriad of intricate limitations and challenges that significantly shape its efficacy and applicability. The quality and composition of the dataset stand as a critical factor in this context. Dataset constraints, encompassing size and diversity, wield substantial influence over sentiment analysis model performance. Constricted datasets might fail to encompass the broad spectrum of sentiments and linguistic intricacies inherent in real-world text data, thereby impeding the models' ability to generalize effectively. Furthermore, the intricacies of language syntax and semantics amplify the complexities. Words frequently embody multiple meanings contingent on their context, potentially leading to misconstructions by models. For example, the term "bad" can project negative sentiment ("It's a bad movie") or positive sentiment ("That's a bad ride!"). This contextual reliance is particularly pronounced in sentiment analysis, necessitating a nuanced grasp of the contextual fabric. Contextual ambiguity emerges as a significant hurdle, as a lone word's sentiment can shift based on its contextual environment. This phenomenon surfaces in phrases like "I'm sick of this" and "That's sick!", wherein the same word garners opposing sentiments. Conquering contextual ambiguity proves vital for precise sentiment classification but poses substantial challenges to models. Additionally, the realms of irony and sarcasm introduce intricacies. These linguistic nuances involve sentiments that veer from literal word meanings. Discerning and comprehending irony and sarcasm necessitate deciphering the speaker's intent and recognizing contextual cues. Models that disregard this complexity risk misclassifying such instances, potentially compromising result accuracy. Imbalanced datasets, where a sentiment class

overshadows others, can inject bias into predictions. Models trained on skewed data might favor the dominant class, resulting in lower accuracy for minority classes. Addressing this imbalance is imperative for balanced sentiment analysis outcomes. Moreover, the struggles extend to domain and context shifts. Models fine-tuned or trained within specific domains might falter in adapting to new contexts, given sentiment variations tied to subject matter, cultural nuances, or geographical distinctions. This underscores the continual need for model adaptation and retraining to sustain accuracy across diverse contexts.

Remarkably, the incorporation of a slang dictionary has emerged as a valuable asset in enhancing sentiment analysis outcomes. Through mapping slang terms to their standard English counterparts, models achieve enhanced proficiency in navigating informal language and discerning intended sentiments accurately. This underscores the utility of integrating a slang dictionary as a preprocessing step in sentiment analysis pipelines. A comparison between SVM and NB results and those of deep learning models unveils that the NB + TF-IDF and LSTM model exhibit the highest accuracy at 0.88, with SVM following at 0.87. The findings from Table 3 and Table 4 lead to the conclusion that LSTM and NB + TF-IDF outperformed machine learning models, SVM and GRU, in terms of accuracy. Among deep learning models, LSTM secured a higher accuracy (0.88) compared to GRU (0.87) within contexts with and without slang dictionaries. Notably, LSTM showcased superior or comparable precision, recall, and F1-scores for both class 0 and class 1, attesting to its prowess in sentiment classification. Conversely, machine learning models generally attained lower accuracy scores, except for NB + TF-IDF.

References

- Aljedaani, W., Rustam, F., Mkaouer, M. W., Ghallab, A., Rupapara, V., Washington, P. B., ... & Ashraf, I. (2022). Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry. *Knowledge-Based Systems*, 255, 109780.
- Baid, P., Gupta, A., & Chaplot, N. (2017). Sentiment analysis of movie reviews using machine learning techniques. *International Journal of Computer Applications*, 179(7), 45-49.
- Fitri, V. A., Andreswari, R., & Hasibuan, M. A. (2019). Sentiment analysis of social media Twitter with case of Anti-LGBT campaign in Indonesia using Naïve Bayes, decision tree, and random forest algorithm. *Procedia Computer Science*, 161, 765-772.
- Harish, B., Kumar, K., & Darshan, H. (2019). Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method. *International Journal of Interactive Multimedia and Artificial Intelligence*.
- Hassonah, M. A., Al-Sayyed, R., Rodan, A., Ala'M, A. Z., Aljarah, I., & Faris, H. (2020). An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter. *Knowledge-Based Systems*, 192, 105353.
- Iswanto, B. H., & Poerwoto, V. (2018, November). Sentiment analysis on Bahasa Indonesia tweets using Unigram models and machine learning techniques. In *IOP Conference Series: Materials Science and Engineering* (Vol. 434, No. 1, p. 012255). IOP Publishing.
- Jalani, M. S., Ng, H., Yap, T. T. V., & Goh, V. T. (2022). Performance of Sentiment Classification on Tweets of Clothing Brands. *Journal of Informatics and Web Engineering*, 1(1), 16-22.
- Kristiyanti, D. A., Kaafi, A. A., Purwaningsih, E., Nurelasari, E., & Nisa, B. (2023, May). Deep learning for Twitter sentiment analysis about the pros and cons of Covid-19 vaccines in Indonesia. In *AIP Conference Proceedings* (Vol. 2714, No. 1). AIP Publishing.
- Matsumoto, K., Yoshida, M., & Kita, K. (2018, September). Classification of emoji categories from

tweet based on deep neural networks. In *Proceedings of the 2nd International Conference on Natural Language Processing and Information Retrieval* (pp. 17-25).

Mussalimun, E. H. (2021). Comparison of K-Nearest Neighbor (K-NN) and Naive Bayes Algorithm for Sentiment Analysis on Google Play Store Textual Reviews. *8th International Conference on Information Technology, Computer and Electrical Engineering*.

Neonardi, S. M. (2021). Aspect Based Sentiment Analysis: Restaurant Online Review Platform in Indonesia with Unsupervised Scraped Corpus in Indonesian Language. *1st International Conference on Computer Science and Artificial Intelligence*.

Pradana, A. W., & Hayaty, M. (2019). The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on indonesian-language texts. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 375-380.

Saad, S. E., & Yang, J. (2019). Twitter Sentiment Analysis Based on Ordinal Regression. *IEEE Access*.

Sairamvinay Vijayraghavan, D. B. (2020). Sentiment Analysis in Drug Reviews using Supervised Machine Learning Algorithms. *arXiv preprint arXiv:2003.11643*.

Sunitha, D. P. (2022). Twitter sentiment analysis using ensemble based deep learning model towards COVID-19 in India and European countries. *Pattern Recognition Letters*, 164-170.

Tyagi, Priyanka, & Tripathi. (2019). A Review towards the Sentiment Analysis Techniques for the Analysis of Twitter Data. *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*.

Venkateswarlu Bonta, N. K. (2019). A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis. *Asian Journal of Computer Science and Technology*.

Yang, S. E. (2019). Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment. *Proceedings of International MultiConference of Engineers and Computer Scientists*.

Yin, C. J., Anawar, S., Othman, N., & Zainudin, N. M. (2021). Slangs and Short forms of Malay Twitter Sentiment Analysis. *IJCSNS International Journal of Computer Science and Network Security*.