JURNAL INFORMATIKA, Vol.12 No.1 April 2025, Page 37-45
ISSN (Print) 2355-6579 | ISSN (Online) 2528-2247
DOI: https://doi.org/10.31294/inf.v12i1.24773

Optimizing Sentiment Analysis on the Linux Desktop Using N-Gram Features

Muhamad Taufiq Hidayat[1*], Rudi Kurniawan[2], Tati Suprapti[3]

[1,2,3] STMIK IKMI Cirebon
Jl. Perjuangan No. 10 B Majasem, Kesambi, Cirebon, Indonesia

Correspondence e-mail: taufiq4hd@gmail.com

Submission:	Revision:	Acceptance:	Available Online:
23-12-2024	05-02-2025	24-02-2025	20-03-2025

Abstract

Linux, or GNU/Linux, is a widely used open-source operating system built on the Linux kernel that is available for anyone to use, known for its security and privacy advantages. With advancements in information technology, protecting privacy has become increasingly challenging due to data extraction practices done by major tech companies. This has encouraged some Mastodon users to switch to Linux, with many expressing their opinions on using Linux as their main operating system. This research seeks to analyze the sentiments of Mastodon users toward Linux through sentiment analysis to understand whether the trend is predominantly positive, negative, or neutral. The methodology used includes collecting data with the help of the Mastodon.py library which then gets manually labelled with the assistance of a linguistic expert as well as a linguistic rule proposed by previous research. The text mining process includes preprocessing steps which includes feature extraction with n-Gram to gain the most optimized result as well as employing feature selection using TF-IDF. The Naïve Bayes algorithm is employed for text classification. The entire process of data analysis is conducted with the help of AI Studio (RapidMiner) software. The results show that the highest-performing model for sentiment analysis is achieved with an n-gram value of 3, revealing user sentiment polarity towards Linux on Mastodon as follows: 42% positive, 28% negative, and 30% neutral. The sentiment analysis model has an accuracy of 63%, with a precision of 70%, recall of 80%, and an f1-score of 74% which shows that this method is able to optimize the sentiment analysis process.

Keywords: n-Gram Feature, Sentiment Analysis, Linux Desktop

1. Introduction

Linux, often referred to as GNU/Linux, is an OS that utilizes the Linux kernel. It is one of several operating systems widely used across various fields, including software development, computer science research, and education (Blum, 2023). Linux is different compared to its commercial counterparts, in that it's free as in its code is licensed with open-source license which means that everyone is allowed to use as well as modify the software to their hearts' content (Boras et al., 2020). Besides that, most Linux distributions that are available are free to download and used with the exceptions of some enterprise Linux distributions. Linux also offers the protection of user privacy as one of the features which has been one of the reasons people use Linux as their main operating system (Bazuku et al., 2023).

Privacy is a fundamental human right. With the increasing advancements of information technology within the 21st century, the methods in which people can apply to protect their own privacy has gone increasingly more difficult to achieve. One of the main causes of this is the ever-evasive methods employed by big tech companies to extract as much data as possible from their own users (Humble, 2021). One recent example of this is Microsoft's Recall which is a tool that works by capturing the user's screen every several minutes which are stored locally within the user's device which then gets analyzed using Artificial Intelligence to extract any available information. The user can then search whatever information that might be available on the saved screenshots. This feature is available for Windows 11 and is not a feature that users can opt-out of which sparked many controversies as well as discourses regarding the usage of Linux as a free and open-source alternative to that of Windows (Kissell, 2024).

Mastodon is a decentralized microblogging social media which operates similarly to X (formerly known as Twitter) and to that of web or email servers (Brembs et al., 2023). Many of Mastodon users have decided to choose Linux as an alternative operating system and many of them

have expressed their opinions which contains varying sentiments regarding Linux. The author could then collect these opinions and use them to analyze the sentiments of Linux as a desktop operating system using sentiment analysis with the aim of discovering valuable information as it pertains to Linux in a form of distributions of positive, negative, as well as neutral opinions.

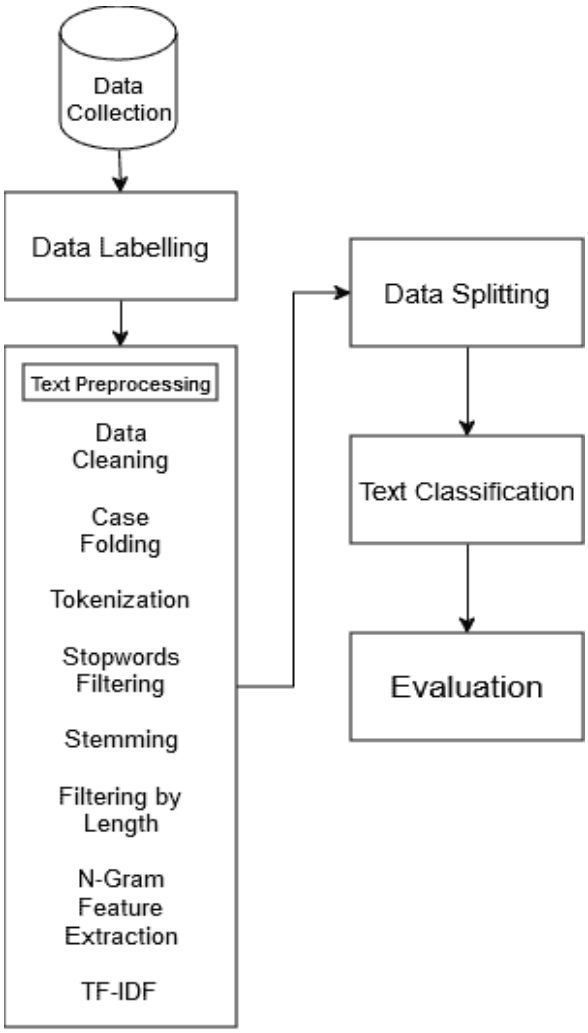
Sentiment analysis is the process of text analysis which includes text mining as well as text classification (Cheng & Chen, 2019). In this research, the author utilizes the n-Gram feature in the preprocessing step which is used to optimize the performance of the analysis sentiment model (Aljameel et al., 2021). The researcher would then compare which implementations of n-Gram feature generates the most optimal performance for the analysis sentiment model. Naïve Bayes algorithm is used to classify the sentiments of the texts. Naïve Bayes is one of the most frequently used classification algorithms that have been used to classify sentiments especially in analyzing texts in the form of snippets, such as Mastodon posts, due to its reliability as well as good performance (Abbas et al., 2019).

Previous study such as (Lazuardi et al., 2023) which only implements the Naïve Bayes classifier shows a moderate performance of the sentiment analysis process with an accuracy of 50%. Other study such as (Tiffani, 2020) which similar method of text classification as well as the implementation of the unigram (n=1) type of n-Gram during the preprocessing step produces analysis sentiment result with an accuracy of 81%, has shown that the performance of a sentiment analysis process can be improved using these two methods.

This study employs n-Gram feature extraction in combination with a Naïve Bayes classifier to identify which n-Gram variant yields the best performance. In addition, it examines sentiment trends on Mastodon regarding Linux as an operating system. The paper is organized as follows: the introduction outlines the research problems and objectives; the methodology section describes the processes and techniques used; the results section discusses the findings in detail; and the conclusion summarizes the research outcomes, relates them to previous studies, and suggests avenues for future work.

2. Research Methods

This research uses multiple steps of methodologies which involve processes starting from data collection, then data labelling with the help of an expert in the English language, text preprocessing, data splitting using the sampling method provided, text classification using the Naïve Bayes method and finally evaluating the results of the classification process with confusion matrix. Figure 1 shows the flow of the methodology that will be carried out throughout this research.



Source: Research Process

Figure 1. Research Methodology

2.1 Data Collection

This process is done in order to collect the data needed for the process of analysis sentiment (Atmadja et al., 2019). The programming language Python is utilized during the process of data collection. The author uses the Python library Mastodon.py for the purpose of interacting with the Mastodon API and crawl through and collect the data needed. The posts or “toot” that are relevant to this research are searched using the keywords “Linux” as well as “Linux Desktop”. The data collection process is done throughout the month of September 2024. The collected data is then saved into the Excel format (xlsx).

2.2 Data Labelling

In this research, sentiment labels consist of positive, negative and neutral labels. The labelling process is done manually using the help of a linguistic expert as well as implementing the

Linguistic Semantic Rules proposed by (Saraswathi et al., 2023).

2.3 Text Preprocessing

Preprocessing or text preprocessing is the process of preparing the data that will be used which includes:

- Data Cleaning
  The first step is removing unwanted elements that may influence or inhibit the analysis process. This includes removing symbols like the at (@) symbol, the hashtag (#) symbol, URLs, numbers, and punctuation marks (Putu et al., 2021).
- Case Folding
  Case folding involves transforming a letter into its lowercase form (Lestandy et al., 2021).
- Tokenization
  Tokenization breaks down each word in a sentence into individual tokens, which are separated by whitespaces (Abdullah & Rusli, 2021).
- Stopwords Filtering
  The AI Studio (RapidMiner) software removes stop words based on the English dictionary. Typical stop words in the English language include terms such as "the," "a," "an," and others.
- Stemming
  Stemming reduces a word to its most fundamental form. This process helps obtain diverse features from the given dataset (Cheng & Chen, 2019).
- Filtering By Length
  Filtering by length is done to further optimize the result of the analysis, the author implements filtering method based on the length of the word. The average length of the English word is around 4.7 characters (Bochkarev et al., 2012). The author uses the number of 4 characters as the minimum length and 25 characters as the maximum length.
- Feature Extraction using N-Gram
  Feature extraction is the process in which [explain feature extraction]. An n-gram is a series of n consecutive elements in a text (Pang et al., 2015).
- Feature Selection using TF-IDF
  TF-IDF is a feature selection method which operates by calculating how frequent a word appears (TF) as well as the frequency of its inverse document (IDF). TF assumes that a term with the highest frequency of occurrence is more important than the ones that are least frequent. However, this presents a limitation in determining which terms are relevant and which ones aren't, which is the reason why IDF is introduced. The IDF formula is shown by formula (1).

$$\mathrm{IDF}(t,d,D) = \log\frac{|D|}{\mathrm{DF}(t,D)} \quad (1)$$

As well as the formula for TF-IDF shown by formula (2).

$$\mathrm{TF-IDF}(t,d,D) = \mathrm{TF}(t,d) * \mathrm{IDF}(t,d,D) \quad (2)$$

With *t* as the term of a given document, *d* as the document itself, as well as *D* as the corpus of text.

2.4 Data Splitting

The data that is going to be analyzed is divided into two types of data which consist of data for training as well as data for testing, using stratified sampling method with a ratio of 80:20. Stratified sampling is a sampling method that works by dividing the overall dataset into chunks or blocks that is determined by a set of specified rules which ensures that the samples are sufficiently representative and contain the structure of the original data (Cao & Shen, 2022).

2.5 Text Classification

Text classification is a classification process involving the process of assigning sets of predefined categories to a given text or document (Xu, 2016). In this research, text classification is performed using the Naïve Bayes classification algorithm which uses the Bayesian theorem as illustrated by equation (3).

$$P(c|x) = P(x|c)\frac{P(c)}{P(x)} \quad (3)$$

With P(c) represents the prior probability of class of c, and P(x) represents the prior probability of the vector x. Previous studies, such as Xia and Yan (2021), have demonstrated that the Naïve Bayes classification method is highly effective for text classification. Its simplicity, relatively high efficiency, and strong stability, particularly when classifying shorter texts like Mastodon posts, contribute to its effectiveness.

2.6 Evaluation

The result of the analysis process is evaluated using the confusion matrix table. As shown by Table 1, confusion matrix is a table consisted of Actual Values as well as Predicted Values that serves to determine the performance of a classification process (Sianipar et al., 2023).

The author will then compute the recall, precision, and F1-score based on the values from the final confusion matrix. Since AI Studio (RapidMiner) provides the accuracy result for the given process, this research will focus on calculating only the recall, precision, and F1-score using formula (4), (5), and (6) respectively.

Table 1. Confusion Matrix

	true positive	true negative	true neutral
pred. positive			
pred. negative			
pred. neutral			

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FNeg} \quad (5)$$

$$f1 - score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (6)$$

TP which stands for True Positives represents how much actual positive and predicted positive, FP, stands for False Positive, shows the amount of falsely predicted positive data, and FNeg represents the amount of falsely predicted negative in a given dataset.

3. Results and Discussion

The processes done throughout this entire research is documented in this section. The author uses Google Colab as well as the programming language Python for collecting the data that are relevant to this research and for the rest of the process the software AI Studio (RapidMiner) is going to be utilized.

3.1 Data Collection

Before collecting data, the author first registered a web application on the settings page of the Mastodon account. This is done with the aim of allowing the usage of the Mastodon API through the utilization of the access token of the successfully created web application. The entirety process of data collection is carried out using Google Colab as well as the Python programming language which involves several libraries including the Mastodon.py library which serves as a method to communicate with the Mastodon API of one of Mastodon's instances which in this research is *toot.community*. The author uses the "Linux" as well as "Desktop Linux" queries to search for relevant posts. This process results in the collection of 960 Mastodon posts which then saved into the Excel format.

3.2 Data Labelling

The data labelling process is done manually with the help of a linguistic expert as well as implementing the *Linguistic Semantic Rules* proposed by (Saraswathi et al., 2023). Table 2 explains each of the rules that will guide the process of data labelling.

Table 2. Semantic Rules

Rule No.	Rules
R1.	For sentences containing the conjunction "but," only the clause following the word "but" is analyzed, while the preceding clause is excluded.
R2.	In sentences with the term "however," the analysis focuses solely on the portion following "however," disregarding the content preceding it.
R3.	When the word "despite" is present, only the clause preceding "despite" is taken into consideration, and any content following it is omitted from analysis.
R4.	If a sentence includes the term "while," the clause following "while" is excluded, and the content after the clause containing "while" is considered instead.

Source: (Saraswathi et al., 2023)

Each data will then be given a label of either Positive, Negative, or Neutral. The result of the data labelling process is shown by Table 3.

Table 3. Labelled Data

Opinion	Label
Linux might not be for everyone, but for me it works great.	Negative
I think Linux Desktop does serves me quite well when I was at school. However, the amount of time it took to just do anything that's beyond some simple web browsing will not fly for my current schedule.	Negative
Well it's safe to say Linux is quite easy to use nowadays despite the apparent lack of other useful apps just like when I first used it years ago.	Positive
while i think it's easy to just use any printers with Linux nowadays, there are still some niche printers out there that are such a pain in the ass to be used with Linux.	Negative

Source: Research Process

3.3 Text Preprocessing

Once all of the data has been given a label, the next phase is text preprocessing. The initial step in this process involves cleaning the data to remove unnecessary elements that could impact the research outcomes. These elements include URLs, numerical values, and punctuation marks. The Replace operator in the AI Studio software is used to carry out the cleaning process. An example of the results from URL cleaning is presented by Table 4.

Table 4. Cleaned Data

Before	After
Share drop - Free, open-source, and cross-platform alternative to AirDrop for Linux. Download here: https://github.com/ShareDropio/sharedr	Share drop Free opensource and cross platform alternative to AirDrop for Linux Download here
TB a turn based battle game against a Magic Dragon i released some time ago! Plays in browser at https://o139.itch.io/tb	TB a turn based battle game against a Magic Dragon i released some time ago Plays in browser at
Finally. After fiddliing around with trying to uncompress the kernel and such in Yggdrasil it was easier to just mount the hard disk under windows and uncompress it there, so I can compile it https://oldbytes.space/tags/Linux	Finally After fiddliing around with trying to uncompress the kernel and such in Yggdrasil it was easier to just mount the hard disk under windows and uncompress it there, so I can compile it
Deploying a Django Project Manually on a Linux Server with uWSGI and Nginx https://www.thedevspace.io/community/django-deploy	Deploying a Django Project Manually on a Linux Server with uWSGI and Nginx

Source: Research Process

The next step is case folding, which involves converting all uppercase letters to lowercase. This process is performed automatically by the AI Studio software with the result shown by Table 5.

Table 5. Case Folding

Before	After
Share drop Free opensource and cross platform alternative to AirDrop for Linux Download here	share drop free opensource and cross platform alternative to airdrop for linux download here
TB a turn based battle game against a Magic Dragon i released some time ago Plays in browser at	tb a turn based battle game against a magic dragon i released some time ago plays in browser at
Finally After fiddliing around with trying to uncompress the kernel and such in Yggdrasil it was easier to just mount the hard disk under windows and uncompress it there, so I can compile it	finally after fiddliing around with trying to uncompress the kernel and such in yggdrasil it was easier to just mount the hard disk under windows and uncompress it there, so i can compile it
Deploying a Django Project Manually on a Linux Server with uWSGI and Nginx	deploying a django project manually on a linux server with uwsgi and nginx

Source: Research Process

Once all text has been converted to lowercase, the next step is tokenization. This process involves splitting the text into individual tokens, using whitespace as the delimiter. The Tokenize operator in AI Studio is used to perform this step, and the results are presented in Table 6.

Table 6. Tokenized Texts

Before	After
share drop free open source and cross platform alternative to airdrop for linux download here	[‘share’, ‘drop’, ‘free’, ‘open’, ‘source’, ‘and’, ‘cross’, ‘platform’, ‘alternative’, ‘to’, ‘airdrop’, ‘for’, ‘linux’, ‘download’, ‘here’]
tb a turn based battle game against a magic dragon i released some time ago plays in browser at	[‘tb’, ‘a’, ‘turn’, ‘based’, ‘battle’, ‘game’, ‘against’, ‘a’, ‘magic’, ‘dragon’, ‘i’, ‘released’, ‘some’, ‘time’, ‘ago’, ‘plays’, ‘in’, ‘browser’, ‘at’]

Before	After
finally after fiddliing around with trying to uncompress the kernel and such in yggdrasil it was easier to just mount the hard disk under windows and uncompress it there so i can compile it	[‘finally’, ‘after’, ‘fiddling’, ‘around’, ‘with’, ‘trying’, ‘to’, ‘uncompress’, ‘the’, ‘kernel’, ‘and’, ‘such’, ‘in’, ‘yggdrasil’, ‘it’, ‘was’, ‘easier’, ‘to’, ‘just’, ‘mount’, ‘the’, ‘hard’, ‘disk’, ‘under’, ‘windows’, ‘and’, ‘uncompress’, ‘it’, ‘there’, ‘so’, ‘i’, ‘can’, ‘compile’, ‘it’]
deploying a django project manually on a linux server with uwsgi and nginx	[‘deploying’, ‘a’, ‘django’, ‘project’, ‘manually’, ‘on’, ‘a’, ‘linux’, ‘server’, ‘with’, ‘uwsgi’, ‘and’, ‘nginx’]

Source: Research Process

After the text has been successfully tokenized, the next step is to filter out stop words from each sentence. This process is conducted using the Filter Stopwords (English) operator provided by the AI Studio software with the removal result shown by Table 7.

Table 7. Stopword Removal

Before	After
share drop free open source and cross platform alternative to airdrop for linux download here	share drop free open source cross platform alternative airdrop linux download here
tb a turn based battle game against a magic dragon i released some time ago plays in browser at	tb turn based battle game magic dragon released time plays browser
finally after fiddliing around with trying to uncompress the kernel and such in yggdrasil it was easier to just mount the hard disk under windows and uncompress it there so i can compile it	finally fiddliing trying uncompress kernel yggdrasil easier mount hard disk under windows uncompress compile
deploying a django project manually on a linux server with uwsgi and nginx	deploying django project manually linux server uwsgi nginx

Source: Research Process

Once all stop words have been removed, the next step is stemming, which involves reducing each word to its root form. This process is performed using the Stemming (Porter) operator provided by the AI Studio software, with the results displayed in Table 8.

Table 8. Stemming

Before	After
share drop free open source cross platform alternative airdrop linux download here	share drop free open source cross platform alternate airdrop linux download here
tb turn based battle game magic dragon released time plays browser	tb turn base battle game magic dragon release time play browser
finally fiddliing trying uncompress kernel yggdrasil easier mount hard disk under windows uncompress compile	final fiddle try compress kernel yggdrasil ease mount hard disk under windows compress compile
deploying django project manually linux server uwsgi nginx	deploy django project manual linux server uwsgi nginx

Source: Research Process

The next step is filtering out words shorter than 4 characters and longer than 25 characters. This is done using the Filter Tokens (by Length) operator in the AI Studio software, with the results shown in Table 9.

Table 9. Filtering based on character length

Before	After
share drop free open source cross platform alternate airdrop linux download here	share drop free open source cross platform alternate airdrop linux download here
tb turn base battle game magic dragon release time play browser	turn base battle game magic dragon release time play browser
final fiddle try compress kernel	final fiddle compress kernel yggdrasil
linux server with uwsgi and nginx	linux server uwsgi nginx

Source: Research Process

Before	After
yggdrasil ease mount hard disk under windows compress compile pre	ease mount hard disk under windows compress compile
deploy django project manual linux server uwsgi nginx	deploy django project manual linux server uwsgi nginx

Source: Research Process

The next and most critical step in preprocessing is generating n-grams using the Generate n-Grams (Terms) operator. In this research, the n-gram types used ranging from unigram (n=1) to trigram (n=3).

The final step of preprocessing is data weighting using TF-IDF which is done using the *Process Documents from Data* operator and setting the prune method of the operator to *absolute*, as well as setting the value 4 for the below absolute pruning threshold and 9999 as its above absolute pruning threshold. The result of TF-IDF is shown by Table 10.

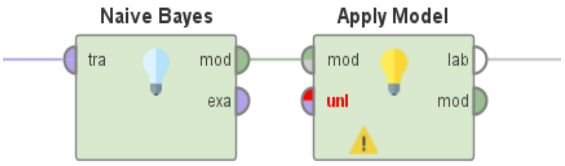
Table 10. Weighted Words

Terms	Weight
fast	0.475
current	0.137
crash	0.517
reliable	0.154
tweak	0.216

Source: Research Process

3.4 Text Classification

After dividing the dataset into two separate sets with the ratio of 80:20, the next step is to implement the Naïve Bayes algorithm and classify the texts. The Naïve Bayes operator, with Laplace correction enabled, is used for this process, along with the Apply Model operator. The workflow of the process of text classification on the AI Studio application is shown in Figure 2.



Source: docs.rapidminer.com

Figure 2. Naive Bayes and Apply Model operator

3.5 Evaluation

The evaluation process is done using the Performance (Classification) operator. The evaluation process occurs three times due to the three types of n-Grams used. The result of each process produces a confusion matrix table.

The process involving unigram produces an accuracy of 62% and the confusion matrix table, presented by Table 11.

Table 11. Confusion Matrix of Unigram

	true positive	true negative	true neutral
pred. positive	49	8	16
pred. negative	13	39	11
pred. neutral	19	6	32

Source: Research Process

In order to calculate the precision, recall as well as f1-score, the author uses the formula (4), (5), and (6) shown by the following calculations.

$$\text{Precision}_{uni} = \frac{TP}{TP + FP} = \frac{49}{49 + 32} = \frac{49}{81} = 0.6 * 100 = 60\%$$

$$\text{Recall}_{uni} = \frac{TP}{TP + FNeg} = \frac{49}{49 + 14} = \frac{49}{63} = 0.77 * 100 = 77\%$$

$$\text{f1} - \text{score}_{uni} = \frac{2 * Recall * Precision}{Recall + Precision}$$

$$= \frac{2 * 0.77 * 0.6}{0.77 + 0.6} = \frac{0.94}{1.38} = 0.68 * 100 = 68\%$$

The process involving the implementation of bigram produces an accuracy of 63% as well as the confusion matrix table shown by Table 12.

Table 12. Confusion Matrix of Bigram

	true positive	true negative	true neutral
pred. positive	54	11	21
pred. negative	12	37	8
pred. neutral	15	5	30

Source: Research Process

The same process of calculating the precision, recall as well as f1-score of the bigram result is shown by the following calculations.

$$\text{Precision}_{bi} = \frac{TP}{TP + FP} = \frac{54}{54 + 27} = \frac{55}{81} = 0.7 * 100 = 70\%$$

$$\text{Recall}_{bi} = \frac{TP}{TP + FNeg} = \frac{54}{54 + 16} = \frac{54}{70} = 0.77 * 100 = 77\%$$

$$\text{f1} - \text{score}_{bi} = \frac{2 * Recall * Precision}{Recall + Precision}$$

$$= \frac{2 * 0.77 * 0.7}{0.77 + 0.7} = \frac{1.08}{1.47} = 0.73 * 100 = 73\%$$

Finally, the process using the trigram type of n-gram generates a classification model with an accuracy of 63%. The corresponding confusion matrix table is presented by Table 13.

Table 13. Confusion Matrix of Trigram

	true positive	true negative	true neutral
pred. positive	55	12	22
pred. negative	12	37	8
pred. neutral	14	4	29

Source: Research Process

The researcher then calculates the value of precision, recall as well as f1-score using the same formula used to calculate the performance of unigram and bigram.

$$\text{Precision}_{tri} = \frac{TP}{TP + FP} = \frac{55}{55 + 26} = \frac{55}{81} = 0.7 * 100 = 70\%$$

$$\text{Recall}_{tri} = \frac{TP}{TP + FNeg} = \frac{55}{55 + 16} = \frac{55}{71} = 0.8 * 100 = 80\%$$

$$\text{f1} - \text{score}_{tri} = \frac{2 * Recall * Precision}{Recall + Precision}$$

$$= \frac{2 * 0.8 * 0.7}{0.8 + 0.7} = \frac{1.12}{1.5} = 0.74 * 100 = 74\%$$

Table 14 shows the comparisons of each performance values produced by three types of n-Grams.

Table 14. Classification performance of three different types n-Grams

N-Gram Type	Accuracy	Precision	Recall	F1-Score
Unigram	62%	60%	77%	68%
Bigram	63%	70%	77%	73%
Trigram	63%	70%	80%	74%

Source: Research Process

According to Table 14 the most optimized performance is obtained when trigram is implemented during the preprocessing step.

4. Conclusion

The experiment results show that the most optimal performance for sentiment analysis on Linux desktops, using data collected from Mastodon, is achieved when the trigram type of n-gram is included during the preprocessing step. The process of analysis sentiment involving trigram results in the distribution of sentiments that includes 42% of opinions with positive sentiments, 28% of opinions with negative sentiments, and 30% of opinions with neutral sentiments. This means that the sentiment on Linux as an operating system on Mastodon leans more toward positive. Furthermore, the performance produced by trigram implementation resulted in sentiment analysis with 63% of accuracy, 70% of precision, 80% of recall, and 74% of f1-score. Compared to the previous study of sentiment analysis only involving the Naïve Bayes method, the result of this experiment outperforms the results obtained by the previous study. This shows that the implementation of n-Gram can improve the performance of a sentiment analysis process.

To obtain higher performance results, there are several things that could be done such as using the data obtained from social media sites with larger userbase than Mastodon such as X, implementing different methods of data labelling using a machine learning model such as VADER, as well as implementing more advanced classification methods such as using a deep learning model.

Reference

Abbas, M., Kamran, A., Memon, Jamali, A. A., Saleemullah Memon, & Anees Ahmed. (2019). *Multinomial Naive Bayes Classification Model for Sentiment Analysis*. Unpublished. https://doi.org/10.13140/RG.2.2.30021.40169

Abdullah, N. A. S., & Rusli, N. I. A. (2021). Multilingual Sentiment Analysis: A Systematic Literature Review. *Pertanika Journal of Science and Technology*, *29*(1). https://doi.org/10.47836/pjst.29.1.25

Aljameel, S. S., Alabbad, D. A., Alzahrani, N. A., Alqarni, S. M., Alamoudi, F. A., Babili, L. M., Aljaafary, S. K., & Alshamrani, F. M. (2021). A Sentiment Analysis Approach to Predict an Individual's Awareness of the Precautionary Procedures to Prevent COVID-19 Outbreaks in Saudi Arabia. *International Journal of Environmental Research and Public Health*, *18*(1), Article 1. https://doi.org/10.3390/ijerph18010218

Atmadja, A. R., Uriawan, W., Pritisen, F., Maylawati, D. S., & Arbain, A. (2019). Comparison of Naive Bayes and K-nearest neighbours for online transportation using

sentiment analysis in social media. *Journal of Physics: Conference Series*, *1402*(7), 077029. https://doi.org/10.1088/1742-6596/1402/7/077029

Bazuku, R., Anab, A., Gyemerah, S., & Mohammed, I. D. (2023). *An Overview of Computer Operating Systems and Emerging Trends* (SSRN Scholarly Paper No. 4609975). Social Science Research Network. https://papers.ssrn.com/abstract=4609975

Blum, R. (2023). *Linux Fundamentals* (2nd ed.). Jones & Bartlett Learning.

Bochkarev, V., Shevlyakova, A., & Solovyev, V. (2012). Average word length dynamics as indicator of cultural changes in society. *Social Evolution and History*, *14*, 153–175.

Boras, M., Balen, J., & Vdovjak, K. (2020). Performance Evaluation of Linux Operating Systems. *2020 International Conference on Smart Systems and Technologies (SST)*, 115–120. https://doi.org/10.1109/SST49455.2020.9264055

Brembs, B., Lenardic, A., Murray-Rust, P., Chan, L., & Irawan, D. E. (2023). Mastodon over Mammon: Towards publicly owned scholarly knowledge. *Royal Society Open Science*, *10*(7), 230207. https://doi.org/10.1098/rsos.230207

Cao, L., & Shen, H. (2022). CSS: Handling imbalanced data by improved clustering with stratified sampling. *Concurrency and Computation: Practice and Experience*, *34*(2), e6071. https://doi.org/10.1002/cpe.6071

Cheng, C.-H., & Chen, H.-H. (2019). Sentimental text mining based on an additional features method for text classification. *PLOS ONE*, *14*(6), e0217591. https://doi.org/10.1371/journal.pone.0217591

Humble, K. P. (2021). International law, surveillance and the protection of privacy. In *The Right to Privacy Revisited*. Routledge.

Kissell, J. (2024). *Take Control of Your Online Privacy, 5th Edition*. alt concepts.

Lazuardi, M. T., Suprapti, T., & Wijaya, Y. A. (2023). Perancangan Model Sentimen Tweet Terhadap Pilkada Dki Jakarta Tahun 2017 Menggunakan Algoritma Naïve Bayes. *Jati (Jurnal Mahasiswa Teknik Informatika)*, *7*(1), Article 1. https://doi.org/10.36040/jati.v7i1.6328

Lestandy, M., Abdurrahim, A., & Syafa'ah, L. (2021). Analisis Sentimen Tweet Vaksin COVID-19 Menggunakan Recurrent Neural Network dan Naïve Bayes. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(4), Article 4. https://doi.org/10.29207/resti.v5i4.3308

Pang, Y., Xue, X., & Namin, A. S. (2015). Predicting Vulnerable Software Components through N-Gram Analysis and Statistical Feature Selection. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 543–548. https://doi.org/10.1109/ICMLA.2015.99

Putu, N. L. P. M., Amrullah, A. Z., & Ismarmiaty. (2021). Analisis Sentimen dan Pemodelan Topik Pariwisata Lombok Menggunakan Algoritma Naive Bayes dan Latent Dirichlet Allocation. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(1), Article 1. https://doi.org/10.29207/resti.v5i1.2587

Saraswathi, N., Sasi Rooba, T., & Chakaravarthi, S. (2023). Improving the accuracy of sentiment analysis using a linguistic rule-based feature selection method in tourism reviews. *Measurement: Sensors*, *29*, 100888. https://doi.org/10.1016/j.measen.2023.100888

Sianipar, J. F., Ramadhan, Y. R., & Jaelani, I. (2023). Analisis Sentimen Pembangunan Kereta Cepat Jakarta-Bandung di Media Sosial Twitter Menggunakan Metode Naive Bayes. *KLIK: Kajian Ilmiah Informatika Dan Komputer*, *4*(1), Article 1. https://doi.org/10.30865/klik.v4i1.1033

Tiffani, I. E. (2020). Optimization of Naïve Bayes Classifier By Implemented Unigram, Bigram, Trigram for Sentiment Analysis of Hotel Review. *Journal of Soft Computing Exploration*, *1*(1), Article 1. https://doi.org/10.52465/joscex.v1i1.4

Xia, X., & Yan, J. (2021). Construction of Music Teaching Evaluation Model Based on Weighted Naïve Bayes. *Scientific Programming*, *2021*(1), 9. https://doi.org/1058-9244

Xu, S. (2016). *Bayesian Naïve Bayes classifiers to text classification*. *44*(1). https://doi.org/10.1177/0165551516677946