A Comparative Study of Text Classification Performance Using NBC and KNN with N-Gram Features in E-Government Services

Istiadi[1]*, Aviv Yuniar Rahman[2], Kuncahyo Setyo Nugroho[3]

[1,2] Widya Gama University, Malang JL Borobudur, No. 35, Mojolangu, Kec. Lowokwaru, Kota Malang, Jawa Timur 65142, Indonesia
[3] Bina Nusantara University, Jl. Araya Mansion No.8 - 22, Genitri, Tirtomoyo, East Java, Indonesia

Keywords

Text Classification; N-Gram Features; Naïve Bayes Classifier (NBC); K-Nearest Neighbor (KNN); E-Government Feedback Analysis

***Correspondence Email:**
istiadi@widyagama.ac.id

Abstract

The Sambat Online service is a promising platform developed by the Malang City Government to collect feedback and suggestions from the public as part of efforts to enhance government service quality. To ensure that citizen feedback is effectively categorized and delivered to the relevant departments, a highly accurate text classification method is required. This study investigates the application of N-Gram features in two popular text classification algorithms: Naïve Bayes Classification (NBC) and K-Nearest Neighbor (KNN). The research aims to analyze and compare the performance of different N-Gram models (unigram, bigram, and trigram), both individually and in combination, for classifying textual data obtained from the Sambat Online system. Experimental results demonstrate that incorporating N-Gram features significantly improves the accuracy of both NBC and KNN classifiers. Among the evaluated methods, NBC achieved consistently higher performance across multiple feature combinations. The highest classification accuracy was obtained when all three N-Gram types were combined, yielding an accuracy of 98.67% for NBC and 97.17% for KNN. These findings indicate that the integration of N-Gram features can effectively enhance text classification performance in e-government applications, particularly when implemented using NBC.

1. Introduction

The increasing adoption of digital public-service platforms has transformed the way citizens interact with governmental institutions. Complaint-reporting systems, in particular, play a pivotal role in enabling governments to receive, assess, and respond to citizen concerns in a timely and effective manner. As public engagement through online mechanisms continues to expand, the volume of user-generated textual complaints has grown substantially, creating challenges for manual categorisation and timely response.

Automating the classification of such complaints has therefore become an essential component of modern e-government infrastructures, particularly in resource-constrained municipalities where human operators are unable to process high traffic efficiently (Bahri et al., 2022).

The Sambat Online platform, established by the municipal government of Malang, Indonesia, exemplifies this trend. Citizens submit textual complaints describing issues related to public services, infrastructure, and administrative matters, each of which must be routed to the appropriate government department for resolution. However, the textual nature of these complaints (which are often short, informal, linguistically diverse, and context-dependent) poses substantial challenges for automated classification. Indonesian-language texts frequently exhibit morphological variations, colloquial expressions, and multi-word constructs whose meanings depend heavily on local context, making simplistic keyword-based approaches insufficient for robust classification (Hadi & Utami, 2024).

To address such challenges, the field of natural language processing (NLP) has increasingly focused on feature engineering strategies capable of capturing deeper contextual information. Among classical approaches, n-gram modelling remains one of the most widely used techniques for representing local linguistic patterns, especially in short-text classification (Kruczek, 2020). N-grams provide a structured means of encoding contiguous token sequences and have been shown to significantly enhance classifier performance in environments where context is crucial for semantic disambiguation (Setiawan et al., 2024).

Alongside feature engineering, algorithm selection plays a critical role in determining classification performance. Classical machine-learning algorithms such as Multinomial Naïve Bayes (NB) and K-Nearest Neighbors (KNN) continue to be effective for text classification due to their interpretability, computational efficiency, and strong performance on sparse, high-dimensional feature spaces (Sheth et al., 2022; Ukey et al., 2023). However, the interaction between feature representation (particularly multi-level n-grams) and algorithmic performance on Indonesian e-government complaint data remains underexplored. Given this gap, the present study investigates how different n-gram configurations influence the performance of NB and KNN for classifying citizen complaints collected from the Sambat Online platform.

1.1 Literature Review

Research on text classification has evolved considerably over the past decade, with feature representation emerging as one of the most influential determinants of model performance. Classical text representations such as bag-of-words (BoW) remain widely used; however, they often fail to capture the syntactic and semantic relationships embedded in contiguous word sequences. N-gram modelling, which encodes token co-occurrence patterns over fixed-length windows, has been repeatedly shown to enhance classification performance particularly in domains characterised by short, context-dependent text segments (Kruczek, 2020). For Indonesian-language NLP, Setiawan et al. (2024) demonstrated that multi-level n-gram TF–IDF representations significantly improve classification accuracy in cyberbullying detection tasks, highlighting the importance of contextual modelling in morphologically complex languages.

Several studies have examined the role of n-grams in domain-specific text analytics. Balande et al. (2024) found that the integration of unigram, bigram, and trigram features consistently outperformed single-gram models in sentiment analysis of hotel reviews, suggesting that multi-gram representation captures both lexical diversity and phrase-level semantics. Chawla et al. (2023) similarly reported that optimising n-gram configurations enhances multilingual short-text classification, with improvements demonstrated across multiple language families. Collectively, these findings reinforce the premise that n-gram combinations are especially valuable for modelling noisy, user-generated content.

In parallel, classical machine-learning algorithms such as NB and KNN continue to serve as strong baselines in text classification due to their simplicity and computational efficiency. NB, which operates under probabilistic assumptions, is particularly well suited for high-dimensional sparse vectors produced by TF–IDF n-gram models (Sheth et al., 2022). Tiffani (2020) showed that NB performs competitively against more complex classifiers in Indonesian text domains when multi-gram features are incorporated. On the other hand, KNN (an instance-based learner) has demonstrated robust performance in high-dimensional settings when paired with cosine similarity, although its performance tends to decline as feature dimensionality increases excessively (Ukey et al., 2023). This behaviour is consistent with known limitations of distance-based classifiers under sparse representations.

Research in the e-government domain remains relatively limited. Bahri et al. (2022) showed that NB outperforms KNN and C4.5 for Indonesian civic complaint classification, although their study did not investigate multi-level n-gram configurations. Nugroho et al. (2020) emphasised the importance of optimising feature selection methods to improve classification outcomes in e-government systems, but the interaction between n-gram representation and algorithmic performance remains insufficiently explored. Despite the global rise in digital governance, only a small number of studies have addressed the unique linguistic and contextual features of citizen-generated complaints.

Taken together, the literature suggests a clear research gap concerning the interplay between n-gram modelling and classical algorithms in the context of Indonesian e-government complaint classification. This study seeks to fill that gap by providing a comprehensive evaluation of unigram, bigram, trigram, and their combinations across NB and KNN models.

2. Research Methods

This research consists of several main processes as shown in Figure 1.

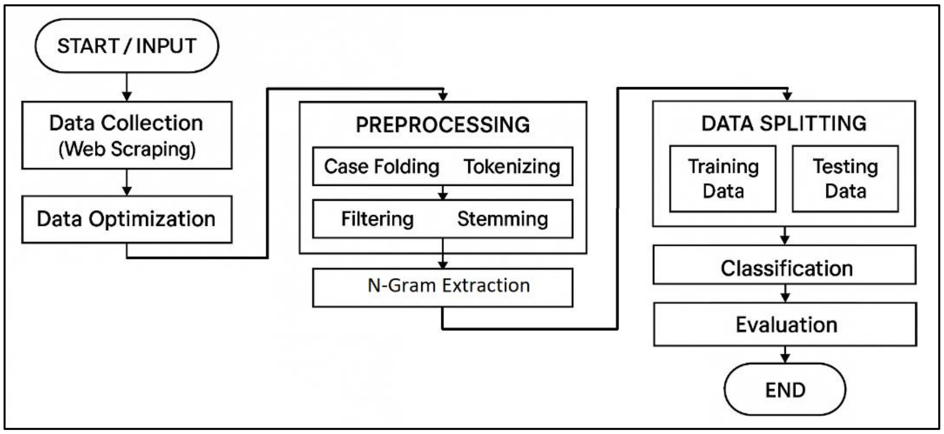


Fig. 1. Research Stages

2.1. Data Collection

The data for this study was taken from the Sambat Online portal using the scraping method, as in the study by Nugroho et al. (2020). The collected data consisted of 350 complaints, each with 13 labels based on the responsible Regional Apparatus Organization (OPD). Not all of the collected data was valid, so validation was required. The resulting data was reduced to 200, divided into 7 classes (labels). These data were then divided into training and test data in .xlxs format (Microsoft Excel) to facilitate further preprocessing. The distribution of the data used is presented in Table 1.

Table 1. Class Data Distribution by OPD

No	OPD Category	Number of Records
1	DISPENDUK	29
2	DLH	25
3	DPUPR	30
4	DISPENDIK	30
5	DISHUB	30
6	DPKP	28

7	SATPOL PP	28
Total Data		**200**

The dataset consists of reports categorized into several regional government agencies (OPD). DISPENDUK refers to the Department of Population and Civil Registration, which manages population administration services. DLH, the Environmental Agency, is responsible for environmental management and public environmental quality monitoring. DPUPR, the Public Works and Spatial Planning Department, oversees public infrastructure development and spatial planning. DISPENDIK, or the Education Department, manages educational services and policies. DISHUB, the Transportation Department, regulates transportation systems and public mobility services. DPKP, the Housing and Settlement Agency, focuses on housing development and residential area management. Finally, SATPOL PP, or the Civil Service Police Unit, is responsible for public order enforcement and regional regulation compliance.

2.2. Data Preprocessing and N-Gram Application

Data preprocessing is performed to prepare the data for processing. Generally, the initial information to be extracted is in an unstructured format, requiring a process of converting it into structured data. This process involves case folding (converting all letters to lowercase), tokenizing (separating each word in a document), filtering (extracting important words from the resulting tokens), and stemming (converting words into their root words).

In this study, the tokenization method used was the n-gram method. N-gram tokenization is divided into two categories: character-pair-based and word-pair-based. In this study, the n-grams used are based on word pairs, truncated from the beginning to the end of the document. N-grams come in various sizes. N-grams with a size of n = 1 are called unigrams, n = 2 are called bigrams, n = 3 are called trigrams, and so on. In this study, only trigrams were used, as more than three would degrade classification performance.

2.3. Feature Selection

Feature selection is the process of selecting a subset of terms that appear in the training data and using this subset as features in text classification to improve accuracy by eliminating noise. The weighting used in this study to select the obtained n-gram features is by using the inverse document frequency (IDF) to assess the importance of a word in a document (Kavadi et al, 2022). The IDF value will be smaller if a term is found in many documents, but will be larger if a term is found in few documents. The IDF for the tth term can be formulated as equation (1).

$$idf_t = log\frac{N}{df_t} \qquad (1)$$

where N is the number of document collections and dft is the number of documents containing term t.

2.4. Classification

Document classification is the activity of placing a document into one or more categories based on its content. In this study, there are seven labels corresponding to the responsible OPD. The general classification function maps a document to a specific category through equation (2).

$$\gamma = X \rightarrow C \qquad (2)$$

where X is a collection of documents and C is a category or class. In this study, classification was performed using NBC and KNN.

2.4.1. Classification using NBC

NBC is a classification method developed based on Bayes' theorem (Nugroho et al, 2020). Multinomial naive Bayes is a development model of the Bayes algorithm suitable for classifying text or documents. In the multinomial naive Bayes formula, document classes are determined not only by the words that appear but also by the number of occurrences. To calculate the probability that word *i* falls into category *j*, we can use equation (3)/

$$P(w_i|c_j) = \frac{count(w_i,c_j)+1}{(\Sigma_{weV} count(w_i,c_j))+|V|} \qquad (3)$$

count(w_i, $_{cj}$) is the number of query words that appear in a class or category. The addition of 1 is used to avoid zero values. *Σ_{weV}count(w_i, c_j)* is the sum of all words in class or category *c_j*. */V/* is the sum of all unique words in all categories.

2.4.2. Classification Using KNN

K-Nearest Neighbors (KNN) is a classification algorithm that uses a distance function between training data to test the data and the number of nearest neighbors to determine the classification results. The distance function used in this experiment is cosine similarity. Cosine similarity is a function widely used in document classification to find similarities between several documents (Bahri et al., 2022). The KNN scoring function is shown in (4). Document class determination is done by voting on the K nearest neighbors. The nearest neighbors are the K documents with the highest similarity scores.

$$score(c, d_1) = \sum_{d_2 \in S_k(d_1)} I_c(d_2) \cos\left(v_{d_1}, v_{d_2}\right) \qquad (4)$$

The *score(c, d_1)* equation is used to measure the degree of relevance between the test document *d_1* and class c by aggregating contributions from its nearest training documents. In this formulation, d_1 denotes the unlabeled test document, while *d_2* represents labeled training documents. Both are expressed as feature vectors (*vd_1* for the test document and *vd_2* for the training document) used to compute similarity. The indicator function *Ic(d_2)* equals 1 if *d_2* belongs to class c and 0 otherwise, ensuring that only training documents relevant to the target class influence the score. The set *S_k(d_1)* contains the k nearest training documents to *d_1*, identified based on similarity measures such as cosine similarity. Consequently, the resulting score reflects the accumulated similarity of the test document to class c through its most similar neighbors.

2.5. Evaluation

The final stage is evaluation, which aims to determine the performance of the classification results for data using the specified classification method. Below are the equations for calculating accuracy (5), precision (6), and recall (7) in a multi-class classification system.

$$Accuracy = \frac{\sum_{i=1}^{l} \frac{TP_i+TN_i}{TP_i+TN_i+FP_i+FN_i}}{l} 100\% \qquad (5)$$

$$Precision = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} (FP_i+TP_i)} 100\% \qquad (6)$$

$$Recall = \frac{\sum_{i=1}^{l} TP_i}{\sum_{i=1}^{l} (TP_i+FN_i)} 100\% \qquad (7)$$

Where *TP_i* is True Positive, which is the number of positive data correctly classified by the system for class i. *TN_i* is True Negative, which is the number of negative data correctly classified by the system for class i. *FN_i* is False Negative, which is the number of negative data but classified incorrectly by the system for class i. *FP_i* is False Positive, which is the number of positive data but classified incorrectly by the system for class i and l is the number of classes.

2.6. Experiment Stage

This experiment uses Rapidminer software, which provides data mining features in the form of operators. The overall configuration of the classifier system is shown in Figure 2.

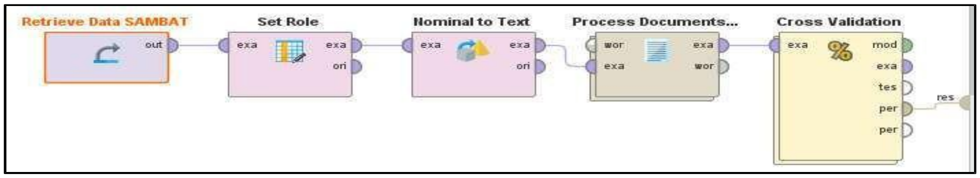


Fig. 2. Classifier System Configuration.

The classifier system consists of the Retrieve Data SAMBAT operator, which loads data from a file into RapidMiner; the Set Role operator, which changes the role of attributes; the Nominal to Text operator, which converts the selected nominal attribute type to text and maps all attribute values to their corresponding string values; the Process Documents from Data operator, which maps words or terms into a vector; and the Cross Validation operator, which performs cross-validation to estimate the statistical performance of the learning model. Furthermore, the Process Documents from Data operator and the Cross Validation operator contain several operators for processing according to their needs. The configuration of the Process Documents from Data operator can be seen in Figure 3.

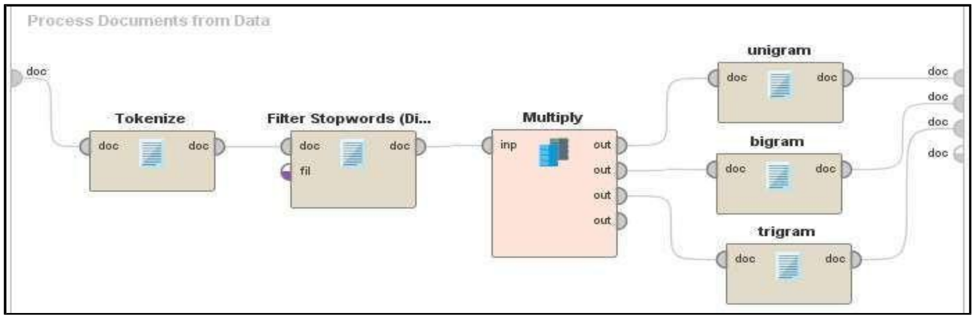


Fig. 3. Configuration of the Process Documents from Data operator

Figure 3 shows the configuration of the operators in the Process Documents from Data operator. The Tokenize operator divides document text into a sequence of tokens. The Stopwords Filter operator removes all tokens that match stopwords from a given file (Dictionary). The Multiply operator multiplies data as needed, in this case for use in multiple n-gram operators.

Meanwhile, the Cross Validation operator contains several operators depending on the machine learning model used. The configuration of the Cross Validation operator using NBC is shown in Figure 4.

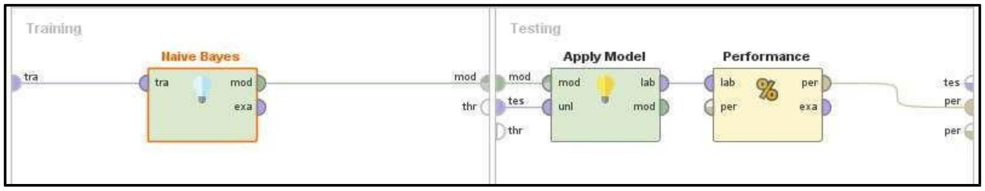


Fig. 4. Configuration of the Cross Validation operator using NBC.

Figure 4 shows the arrangement of the operators in the Cross Validation operator using NBC. The Naïve Bayes operator generates a Naïve Bayes (NBC)-based classification model for training. In the testing section, the Apply Model operator applies the test data set (Example Set) to the NBC model. The Performance operator is used to statistically evaluate classification performance. The configuration of the Cross Validation operator using KNN can be seen in Figure 5.

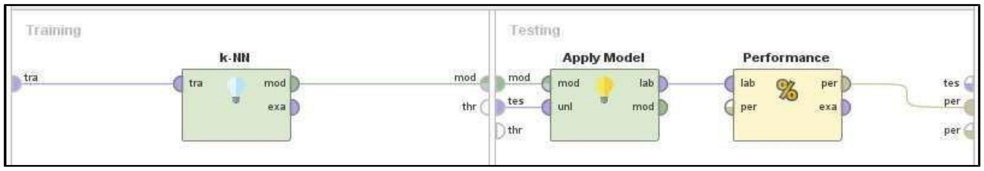


Fig. 5. Configuration of the Cross Validation operator using KNN.

Figure 5 shows the arrangement of operators in the Cross Validation Operator using KNN. The k-NN operator generates a KNN-based classification model in the training section. Meanwhile, in the testing section, there is the Apply Model operator, which applies the test data set (Example Set) to the KNN model. The Performance operator is used to statistically evaluate classification performance.

3. Result and Discussion

Based on the overall configuration, tests were conducted on the NBC and KNN models with various n-gram combinations: unigram (n=1); bigram (n=2); trigram (n=3); unigram and bigram; unigram and trigram; bigram and trigram; unigram, bigram, and trigram. Performance measurements were performed using 10-fold cross-validation. The results of these tests yielded a Confusion Matrix, which then determined the accuracy, recall, and precision values.

In the first stage, experiments were conducted on NBC. The experiments were conducted by applying a predetermined configuration with n-gram variations and a validation method using k-fold cross-validation. The performance of each variation is presented through accuracy, precision, and recall. Table 2 displays the NBC performance of each n-gram variation.

Table 2. Summary of NBC performance based on n-gram variations

n-gram variation	NBC (%)		
	Recall	Precision	Accuracy
Unigram	59.62	58.68	60.00
Bigram	63.02	62.27	63.50
Trigram	63.02	61.74	63.00
Unigram + Bigram	96.23	96.23	96.25
Unigram + Trigram	94.99	94.97	95.00
Bigram + Trigram	98.00	97.95	98.00
Unigram + Bigram + Trigram	98.68	98.63	98.67

Based on Table 2, the evaluation of the Naïve Bayes Classifier (NBC) across various n-gram configurations shows that single-feature n-grams (unigram, bigram, trigram) provide limited discriminative capability, yielding accuracy levels between 60% and 63.5%. However, performance increases substantially when multiple n-gram types are combined. The combinations of unigram + bigram and unigram + trigram improve accuracy to around 95–96%, while bigram + trigram yields even stronger results at 98.00%. The highest performance is achieved when all three n-gram types are integrated simultaneously, enabling NBC to reach 98.67% accuracy, demonstrating that multi-level n-gram fusion provides richer contextual representation and significantly enhances classification effectiveness.

In the second stage, experiments were conducted on KNN. As in the previous experiment, a predetermined configuration with n-gram variations and a k-fold cross-validation validation method were applied. The performance of each variation is presented through accuracy, precision, and recall. For classification using KNN, k with a value of 3 was used as the most optimal parameter. Table 3 shows the KNN performance for each n-gram variation.

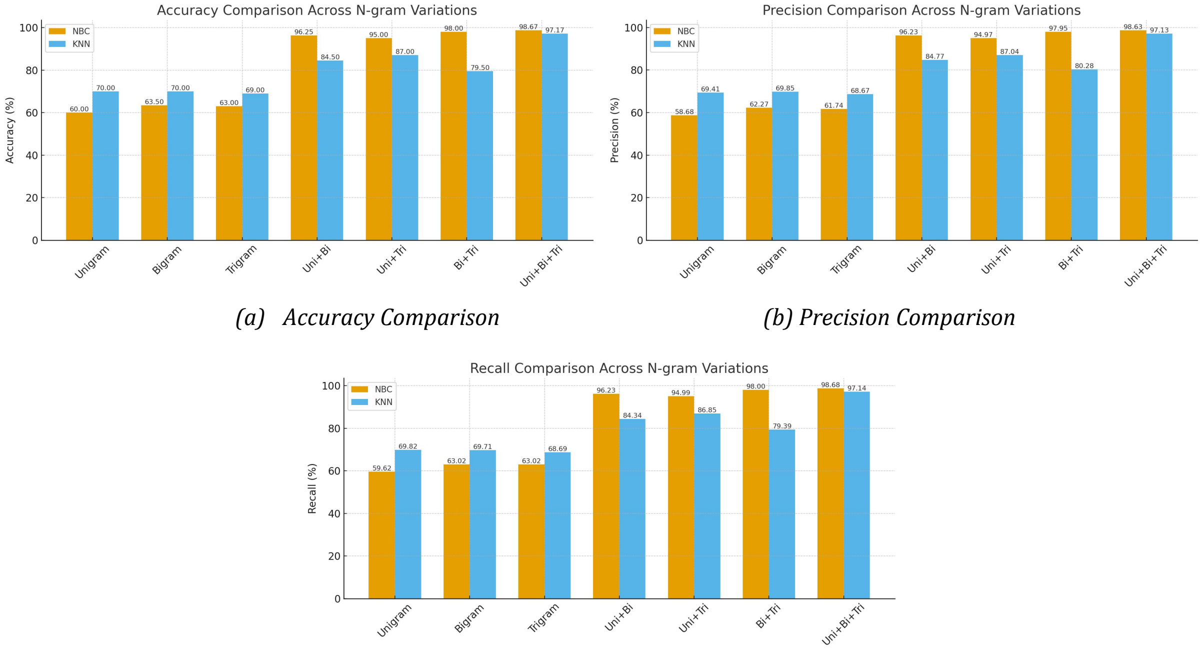
Table 3. Summary of KNN classifier performance based on n-gram variations.

n-gram variation	KNN (%)		
	Recall	Precision	Accuracy
Unigram	69.82	69.41	70.00
Bigram	69.71	69.85	70.00
Trigram	68.69	68.67	69.00
Unigram + Bigram	84.34	84.77	84.50
Unigram + Trigram	86.85	87.04	87.00

Bigram + Trigram	79.39	80.28	79.50
Unigram + Bigram + Trigram	97.14	97.13	97.17

Based on Table 3, the performance of the K-Nearest Neighbor (KNN) classifier varies notably across different n-gram configurations. When using single n-gram features (unigram, bigram, trigram), the classifier achieves moderate accuracy levels between **69% and 70%**, indicating limited discriminative strength. Classification performance improves substantially with combined n-gram features: the **unigram + bigram** configuration reaches **84.50% accuracy**, while **unigram + trigram** yields even better results at **87.00%**, marking the strongest performance among two-way combinations. In contrast, the **bigram + trigram** combination provides a lower improvement of **79.50% accuracy**. The highest performance is obtained when all three n-gram types are integrated simultaneously, enabling KNN to achieve **97.17% accuracy**, demonstrating that multi-level n-gram fusion significantly enhances the classifier's ability to capture contextual patterns and improve text categorization effectiveness.

Based on the overall testing, a comparison was made between the classifier performance using various n-gram combinations. Figure 6 presents a summary of the overall comparion of test results.



(a) Accuracy Comparison

(b) Precision Comparison

(c) Recall Comparison

Fig. 6. Comparison of NBC and KNN performance on N-gram variations (a) Accuracy Comparison (b) Precision Comparison (c) Recall Comparison

The experimental results reveal substantial differences in classifier performance across the various n-gram configurations. Figure-based evaluations demonstrated that single-level n-gram features (unigram, bigram, and trigram) produce only moderate performance for both algorithms. Specifically, Multinomial Naïve Bayes achieved accuracy scores ranging from approximately 60% to 63%, while KNN reached slightly higher values, between 69% and 70%. These results are consistent with prior research indicating that single-token representations often fail to capture the nuanced linguistic patterns necessary for accurately classifying short and informal texts such as those common in citizen complaints (Kruczek, 2020; Hadi & Utami, 2024). Indonesian-language texts, which often contain compound expressions and context-bearing multi-word segments, particularly benefit from richer contextual encoding (Setiawan et al., 2024). Thus, the modest performance of single n-grams aligns with well-established limitations of minimal-context feature spaces.

Performance improved markedly when multi-gram feature combinations were used. For Naïve Bayes, the integration of two n-gram levels (such as unigram+bigram or bigram+trigram) resulted in accuracy scores between 95% and 98%. These gains confirm earlier findings by Balande et al. (2024), who observed that multi-gram representations enable classifiers to capture both lexical variety and local context, resulting in more discriminative feature spaces. KNN also benefited from multi-gram combinations, achieving accuracy values ranging from 79% to 87%. However, KNN's improvements, while substantial, were consistently lower than those of Naïve Bayes. This discrepancy is likely attributable to the distance-based nature of KNN, which becomes increasingly sensitive to feature sparsity as dimensionality increases (Ukey et al., 2023). The expanded n-gram space produces highly sparse vectors, and Naïve Bayes (being a probabilistic classifier) tends to handle such sparsity more effectively (Sheth et al., 2022).

The most significant findings emerged from the three-level n-gram combination (unigram+bigram+trigram), which delivered the highest performance for both algorithms. Naïve Bayes achieved an accuracy of 98.67%, while KNN reached 97.17%. These near-optimal classification results suggest that a comprehensive n-gram representation provides a balanced combination of lexical coverage and contextual specificity. Such a feature space captures both general word usage patterns (via unigrams) and phrase-level semantics (via bigrams and trigrams), enabling more precise assignment of complaints to their corresponding government departments. Similar observations have been reported in multilingual sentiment analysis and hate-speech classification studies, where the combination of multiple contextual levels significantly enhances classifier performance (Chawla et al., 2023; Taha, 2024).

A deeper analysis of the confusion matrices reveals additional insights regarding class-specific behaviour. In particular, classes associated with semantically similar departmental responsibilities exhibited higher misclassification rates when using single-level n-grams. This suggests that isolated lexical cues are insufficient to differentiate nuanced complaint categories. However, with the introduction of multi-gram features, these class-level misclassifications diminished considerably. This pattern is consistent with findings by Rogers et al. (2021), who demonstrated that higher-order n-grams reduce ambiguity by capturing discriminative collocations unique to specific topics or contexts.

The comparative performance between Naïve Bayes and KNN also warrants discussion. Across all n-gram settings, Naïve Bayes consistently outperformed KNN, particularly in high-dimensional multi-gram configurations. This trend aligns with earlier research demonstrating that Naïve Bayes remains a highly competitive baseline for sparse vector representations due to its generative modeling assumptions and computational efficiency (Tiffani, 2020; Sinha et al, 2022). Conversely, KNN's reliance on similarity calculations makes it more vulnerable to the "curse of dimensionality," resulting in diminishing returns when feature dimensionality increases sharply. Nevertheless, KNN's strong performance in the full multi-gram configuration indicates that the enriched feature space successfully captures meaningful semantic neighbourhoods, despite its sensitivity to sparsity.

Taken together, these findings reinforce the central importance of feature engineering in text-classification systems for e-government applications. The dramatic improvements observed with multi-gram models demonstrate that carefully constructed n-gram combinations can significantly enhance classifier effectiveness, even without employing more complex deep-learning architectures. This is particularly relevant for government agencies operating with limited computational resources, where classical models such as Naïve Bayes offer a favourable balance between performance, interpretability, and computational cost.

4. Conclusions

This study demonstrates that feature engineering (particularly through multi-level n-gram modelling) plays a pivotal role in achieving high accuracy in e-government complaint classification. While single-level n-grams capture limited context, combining unigram, bigram, and trigram features substantially enhances performance for both Naïve Bayes and KNN. NB achieved the highest accuracy at 98.67%, with KNN following at 97.17%, illustrating the viability of classical models when paired with rich lexical-contextual features.

The findings affirm that optimised classical approaches can deliver near state-of-the-art performance without the computational overhead of deep-learning architectures, making them well-suited for public-sector environments. Future work should investigate contextual embeddings (e.g., BERT, IndoBERT), explore domain adaptation across municipalities, and evaluate larger multi-source datasets to enhance generalisability.

5. References

Bahri, S., Utami, E., & Nasiri, A. (2022). Classification of public complaints based on text mining using modified K-nearest neighbor, naïve bayes and C4.5 algorithm. CCIT Journal (Creative Communication and Innovative Technology), 15(2), 198–208. https://doi.org/10.33050/ccit.v15i2.2286

Balande, B. B., Kolte, D. M., Manza, R. R., & Revate, S. S. (2024). Literature review on N-gram text classification models for hotel reviews sentiment analysis. In Algorithms for intelligent systems: Proceedings of International Conference on Computational Intelligence (pp. 641-655). Springer Nature Singapore. https://doi.org/10.1007/978-981-97-3526-6_49

Chawla, S., Kaur, R., & Aggarwal, P. (2023). Text classification framework for short text based on TFIDF-FastText. Multimedia Tools and Applications, 82(26), 40167-40180. https://doi.org/10.1007/s11042-023-15211-5

Hadi, K., & Utami, E. (2024). Analysis of K-NN with the Integration of Bag of Words, TF-IDF, and N-Grams for Hate Speech Classification on Twitter. JUITA: Jurnal Informatika, 12(2), 289-298. https://doi.org/10.30595/juita.v12i2.23829

Kavadi, D. P., Sanaboina, C. S., Patan, R., & Gandomi, A. (2022, April). N-Gram-Based Machine Learning Approach for Bot or Human Detection from Text Messages. In Proceedings of the 2022 6th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (pp. 80-85). https://doi.org/10.1145/3533050.3533063

Kruczek, J. (2020). Are n-gram categories helpful in text classification? PLOS ONE, 15(3), e0229151. https://doi.org/10.1007/978-3-030-50417-5_39

Nugroho, K. S., Istiadi, I., & Marisa, F. (2020). Optimasi naive Bayes classifier untuk klasifikasi teks pada e-government menggunakan particle swarm optimization. Jurnal Teknologi dan Sistem Komputer, 8(1), 21-26. https://doi.org/10.14710/jtsiskom.8.1.2020.21-26

Rogers, D., Preece, A., Innes, M., & Spasić, I. (2021). Real-time text classification of user-generated content on social media: Systematic review. IEEE Transactions on Computational Social Systems, 9(4), 1154-1166. https://doi.org/10.1109/TCSS.2021.3120138

Setiawan, Y., Maulidevi, N. U., & Surendro, K. (2024). The Optimization of n-Gram Feature Extraction Based on Term Occurrence for Cyberbullying Classification. Data Science Journal, 23(1). https://doi.org/10.5334/dsj-2024-031

Sheth, V., Tripathi, U., & Sharma, A. (2022). A comparative analysis of machine learning algorithms for classification purpose. Procedia Computer Science, 215, 422-431. https://doi.org/10.1016/j.procs.2022.12.044

Sinha, A., Naskar, M. N. B., Pandey, M., & Rautaray, S. S. (2022, December). Text classification using machine learning techniques: comparative analysis. In 2022 OITS International Conference on Information Technology (OCIT) (pp. 102-107). IEEE. https://doi.org/10.1109/OCIT56763.2022.00029

Taha, K., Yoo, P. D., Yeun, C., Homouz, D., & Taha, A. (2024). A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights. Computer Science Review, 54, 100664. https://doi.org/10.1016/j.cosrev.2024.100664

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), 1-7. https://doi.org/10.52465/joscex.v1i1.4

Ukey, N., Yang, Z., Li, B., Zhang, G., Hu, Y., & Zhang, W. (2023). Survey on exact knn queries over high-dimensional data space. Sensors, 23(2), 629. https://doi.org/10.3390/s23020629