INTENSIF, Vol.10 No.1 2026
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

Word Stemming of Lampung Dialect *Nyo* using N-Gram Stemming

Received:
9 June 2025
Accepted:
1 January 2026
Published:
1 February 2026

[1]Parjito, [2*]Zaenal Abidin, [3]Akmal Junaidi, [4]Wamiliana, [5]Favorisen R. Lumbanraja, [6]Farida Ariyani
[1,2]Sistem Informasi, FTIK, Universitas Teknokrat Indonesia
[3,5]Ilmu Komputer, FMIPA, Universitas Lampung
[4]Matematika, FMIPA, Universitas Lampung
[6]Magister Pendidikan Bahasa dan Kebudayaan Lampung, Universitas Lampung
E-mail: [1]djito@teknokrat.ac.id, [2]zabin@teknokrat.ac.id, [3]akmal.junaidi@fmipa.unila.ac.id, [4]wamiliana.1963@fmipa.unila.ac.id, [5]favorisen.lumbanraja@fmipa.unila.ac.id [6]farida.ariyani@fkip.unila.ac.id
*Corresponding Author

Abstract— Background: Previous translation systems for the Lampung dialect of *nyo* to Indonesian achieved bilingual evaluation understudy (BLEU) scores below 40%, primarily due to challenges in processing affixed words. **Objective**: This research aims to perform stemming on affixed words in the Lampung dialect of *nyo* to enhance the performance of the translation system. **Methods**: We developed an n-gram stemming approach that reduces affixed words to their base forms by measuring similarity between n-grams using the Dice coefficient method. When similarity exceeds a specified threshold, the system identifies the corresponding base word. **Results**: Using a dataset of 700 words from the Lampung dialect of *nyo*, we constructed a comprehensive stemmer covering all affix variations. The optimal threshold was determined to be 0.5, achieving bigram accuracy of 93.86% and trigram accuracy of 89.14%. These accuracy levels demonstrate the method's effectiveness in identifying base word forms, which directly impacts translation quality improvement. **Conclusion**: N-gram stemming with a 0.5 threshold effectively processes the Lampung dialect of *nyo* morphology and shows potential for enhancing translation accuracy. This work represents the first comprehensive stemming system specifically designed for the Lampung dialect of *nyo*, contributing to the development of natural language processing tools for underrepresented regional languages in Indonesia.

Keywords— Stemming; Dialect of *nyo*; N-Gram Stemming; Threshold; Translation

Corresponding Author:
Zaenal Abidin,
Department of Information System,
Universitas Teknokrat Indonesia,
Email : zabin@teknokrat.ac.id
Orchid ID: https://orcid.org/0000-0003-4237-7167

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

I. INTRODUCTION

Lampung dialect of *nyo* is one of the main dialects of the Lampung language spoken in the northern, central and eastern parts of Lampung Province, Indonesia, distinguished from the more common inland dialect of *api* by its characteristic vocabulary and intonation, with "*nyo*" meaning "what" compared to "*api*" used in similar questions, and while it has a traditional script called *Had Lampung*, it is now more typically written with Latin letters in everyday life [1]. As one of the regional languages of the archipelago that speakers still preserve, Lampung serves as an intra-tribal lingua franca and is the language of instruction in traditional ceremonies, such as weddings, naming, granting titles, and circumcision [2]. The Lampung dialect of *nyo* also has potential for exploration from the perspective of natural language processing, particularly in terms of stemming [3], [4].

Stemming is a method that links morphological variants of words to their base forms, commonly used as preprocessing in natural language processing, information retrieval, and language modeling. Singh and Gupta provided a comprehensive examination of text-stemming theory, methodologies, and applications, analyzing significant literature and categorizing prominent stemming algorithms using standard datasets, while describing current state-of-the-art and outstanding concerns in unsupervised stemming [5], [6]. However, numerous performance evaluation standards exist for stemmers, each examining performance from particular standpoints, with research revealing that current evaluation measures may only score average word conflation independent of stem accuracy, often preferentially benefiting specific language varieties like Urdu, and no existing evaluation metrics can effectively assess stemmer performance across all language types [7].

Research on stemming and lemmatization for local languages in Indonesia has begun, starting with research on Indonesian [8], [9]. This research is based on morphological analysis and the use of local language dictionaries, including Javanese [10], [11], [12], Balinese [13], [14], [15], [16], [17], Madurese [18], [19], [20], Sundanese [21], [22], [23], [24], Rejang [25], [26], Minangkabau [27], [28], Riau Malay [29], Lampung [30], [31], and Tetun [32]. Several SLR publications related to stemming and lemmatization are also available to help readers understand state-of-the-art (SOTA) research in this area [3], [33]. The state-of-the-art in stemming research on Lampung language words utilizes brute-force [30] and Rule-based [31] methods. The main principle in brute-force is matching the test word with the base word provided in the database, while rule-based is using all morphological rules in the Lampung language.

Important findings and research gaps that were successfully extracted from the review of several previous stemming studies on local languages in Indonesia are (1) the main reference method of stemming is the research results of Nazief-Adriani stemming, (2) the methods modified by stemming researchers are Nazief-Adriani, Confix-Stripping, Enhanced Confix-Stripping, Rule-based and N-Gram stemming methods, (3) in Lampung language using Brute-force and Rule-based approach. This study represents a first step in overcoming the shortcomings of previous research [34], which has limitations in translating affixed words. The sample cases are given in Table 1.

Table 1. Direct Machine Translation (DMT) Test Sentence Sample [34]

Test Sentences in Dialect of *Nyo*	Result of DMT Without Stemming	Result of DMT With Stemming
menak mejeng di keresei	*paman duduk di kursi*	*paman duduk di kursi*
apui enok ***ngebalak***	*api itu* ***ngebalak***	*api itu besar*

The information based on Table 1 is as follows: (1) Column one contains two sentences in the Lampung language; column two provides their translations without stemming, and column three provides their translations with stemming. (2) The first sentence does not contain any affixed words, while the second sentence contains one affixed word, which is "*nge* - *balak*."

This research developed a stemmer for the Lampung dialect of *nyo* dialect using the n-gram stemming method, which can handle non-rule-recognized affixed words, including writing errors [14].

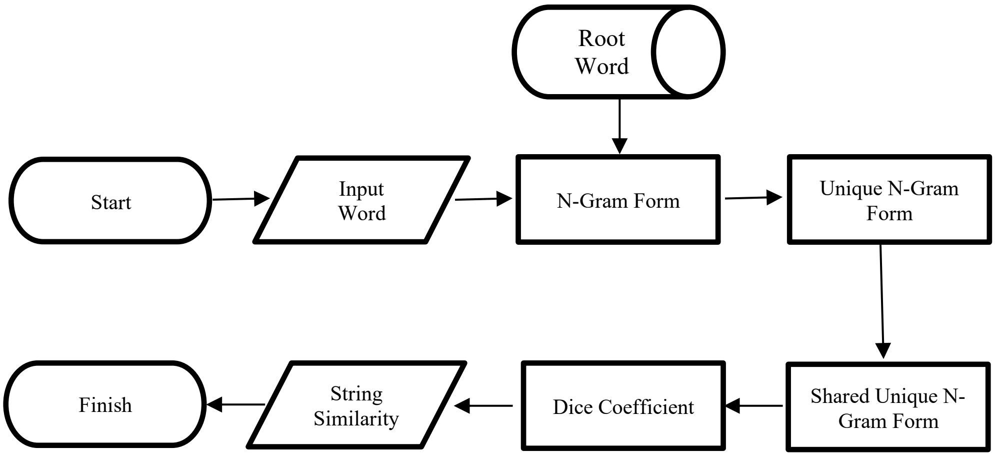

Fig 1. The Flow Process of N-Gram Stemming [14]

N-Gram stemming is used when standard rules fail to recognize affixed words, particularly those with spelling errors or irregularities. The method converts both affixed words and root words into n-gram format, then compares them by counting unique n-grams and shared n-grams between the two words. The similarity is calculated using the Dice coefficient approach as shown in equation (1) [14].

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

$$dc = (2 \, . \, c) / (a + b) \qquad (1)$$

In equation (1), the Dice coefficient *dc* measures similarity between input and root words, where *c* represents shared unique n-grams, *a* indicates input word n-grams, and *b* indicates root word n-grams [14]. The n-gram stemming process utilizes bi-grams (n=2) or tri-grams (n=3) to assess word similarity, presenting root words when computation results exceed threshold values of 0.70, 0.65, or 0.60 [14]. Research testing threshold values between 0.5 and 0.8 found optimal results at 0.5.

II. RESEARCH METHOD

The research stages define the sequential steps conducted in the research process to attain the expected objectives. The steps done are (1) data collection, (2) algorithm design, (3) implementation and testing, and (4) evaluation of result. The research stages are represented in Fig. 2.

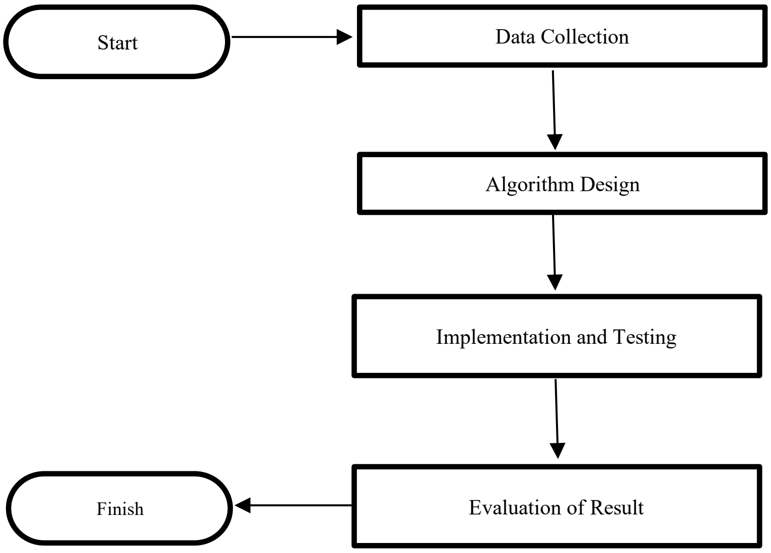

Fig 2. The Research Stages

The stages of research about the N-Gram Stemming stemmer in the dialect of *nyo* are as follows, and pseudocode, and experimental results can be accessed at the link bit.ly/NgramStemming25 :

1. *Data Collection* : This study gathers data using essential word dictionary data, test word data, and independent test word data. Manually inputting 6454 terms into the Lampung-Indonesian Dictionary produced the fundamental lexical information. The dataset utilized for evaluating the N-Gram stemming comprises 700 test words, as referenced in the book *Sistem Morfologi Verba Bahasa Lampung Dialek Tulang bawang* [1].
2. *Algorithm Design*: The formulation of the algorithm is based on the N-Gram stemming algorithm, which relates to how N-Gram Stemming works. The design phase employs

detailed pseudocode. The pseudocode for the N-Gram Stemming *dialect of nyo* is presented at the following link: bit.ly/NgramStemming25.

3. *Implementation and Testing*: Implementation refers to the application of the algorithm derived from the preceding stage. The pseudocode implementation was executed in Python on Google Colab.
4. *Evaluation of result*: Stemming evaluation employing gold standard assessment is a technique for gauging the quality and precision of stemming algorithms by contrasting the algorithm's stemmed outputs with a compilation of accurate base words (gold standard).

A. Details of the Technical Stages of N-Gram Stemming

N-gram stemming is a natural language processing technique comprising five systematic steps to reduce words to their root forms.

1. The process begins with data preparation, including collecting words to be stemmed, establishing a basic word dictionary as a reference, and setting a threshold value of 0.5 as a similarity benchmark.
2. The second step employs the bi-gram method (n=2) by breaking words into pairs of two consecutive characters, doing the same for dictionary words, and then calculating similarity using Dice's Coefficient formula, where words are considered similar if their similarity value exceeds 0.5.
3. Concurrently, the third step applies the tri-gram process (n=3) that segments words into sets of three consecutive characters compares them with dictionary words using the same method, and verifies whether the similarity value exceeds the 0.5 thresholds.
4. Next, the decision-making step selects the base word with the highest similarity value (exceeding 0.5) from either the bi-gram or tri-gram method. The system chooses the shortest if several words share the same similarity value.

Finally, the stemming result is determined by transforming words with similarity values greater than 0.5 into their root forms according to the dictionary reference, while words with similarity values less than or equal to 0.5 remain in their original form, making the entire process an effective method for normalizing morphological variations of words in natural language processing applications.

This process helps identify words with structural similarities and can be used to more accurately find the base word of a compound word. The stemming accuracy results are computed using equation (2) [7].

$$\textit{Gold standar assessment} = (t / m) \times 100 \qquad (2)$$

In equation (2), the gold standard assessment represents the stemming accuracy. *t* represents the quantity of words that have been accurately stemmed, and *m* represents the quantity of fastened words [7].

B. Details of the Example of N-Gram Stemming

The following is a modification of the N-Gram stemming algorithm for Lampung language words in the dialect of *nyo* :

1. Input Data

   Enter the word to be stemmed, convert the word into lowercase letters, determine the value of N (the desired N-gram size), and prepare a dictionary of basic words for comparison.
2. Padding

   Add an underscore character (_) at the beginning of the word, add an underscore character (_) at the end, and save the word that has been padded.
3. N-gram Generation
   a. Start from the first character of the word that has been padded,
   b. Take N consecutive characters,
   c. save the N-character chunks as one N-gram,
   d. Shift one character to the right
   e. Repeat steps b-d until reaching the end of the word
   f. Collect all the N-grams formed
4. Comparison with Base Word
   a. Take one word from the base word dictionary
   b. Perform the padding process on the word from the dictionary
   c. Create an N-gram from the dictionary word
   d. Count the number of N-grams that are the same (intersection)
   e. Count the total N-grams of the two words
   f. Calculate the similarity value with the Dice formula:
      Similarity = (2 × number of same N-grams) / (total N-grams of word1 + total N-grams of word2)
   g. Save the similarity value and base word
   h. Repeat steps a-g for all words in the dictionary
5. Result Selection
   a. Determine the minimum threshold value (e.g. 0.5)
   b. Check all calculated similarity values

   c. Select the root word with the highest similarity value
   d. Compare the highest similarity value with the threshold
   e. If the similarity value > threshold:
      Use the selected root word as the result
   f. If similarity value ≤ threshold:
      Use the original word as the result

6. Result/Output
   a. Display the stemming result word
   b. Show similarity value (optional)
   c. Done

Example:

The word to be stemmed is "pamelok", N = 2 (bi-gram), threshold = 0.5

Padding: "_pamelok_"

Bi-gram: ["_p", "pa", "am", "me", "el", "lo", "ok", "k_"]

Compare with the root word "melok": Base word with padding: "melok"

Base word bi-gram: ["_m", "me", "el", "lo", "ok", "k_"]

N-gram equal: ["me", "el", "lo", "ok", "k_"]

Similarity = $(2 \times 5) / (8 + 6) = 10/14= 5/7 = 0.71$

If 0.71 > threshold, then the stemming result is "melok"

The word to be stemmed is "pamelok", N = 3 (trigram), threshold = 0.5

Padding: "_pamelok_"

Trigram: ["_pa", "pam", "ame", "mel", "elo", "lok", "ok_"]

Compare with the root word "melok": Base word with padding: "melok"

Base word trigram: ["_me", "mel", "elo", "lok", "ok_"]

N-gram equal: ["mel", "elo", "lok", "ok_"]

Similarity = $(2 \times 4) / (7 + 5) = 8/12= 0.67$

If 0.67 > threshold, then the stemming result is "melok"

III. RESULT AND DISCUSSION

In this study, we focus on N-Gram stemming. Our next research will focus on stemming the Lampung dialect of *nyo* using modified Nazief-Adriani, Confix-Stripping, Enhanced Confix-Stripping or Rule-based methods. N-Gram stemming is developed without going through an exploration based on morphological principles, but using a language dictionary. Experiments focused on the environments (1) bigram with threshold 0.5 and 0.55, and (2) trigram with threshold 0.5 and 0.55. The Python code was then tested on the Google Colab environment for

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

implementation. The input consists of 700 test words. All proofs of this experiment are archived and can be accessed at the address bit.ly/NgramStemming25. Figures 3 and 4 summarise the experimental results of N-Gram Stemming.

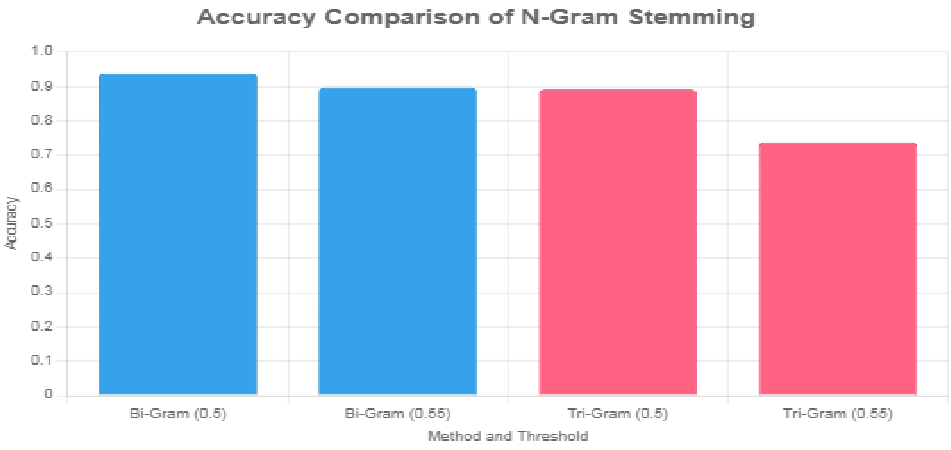

Fig 3. Accuracy Comparison of N-Gram Stemming

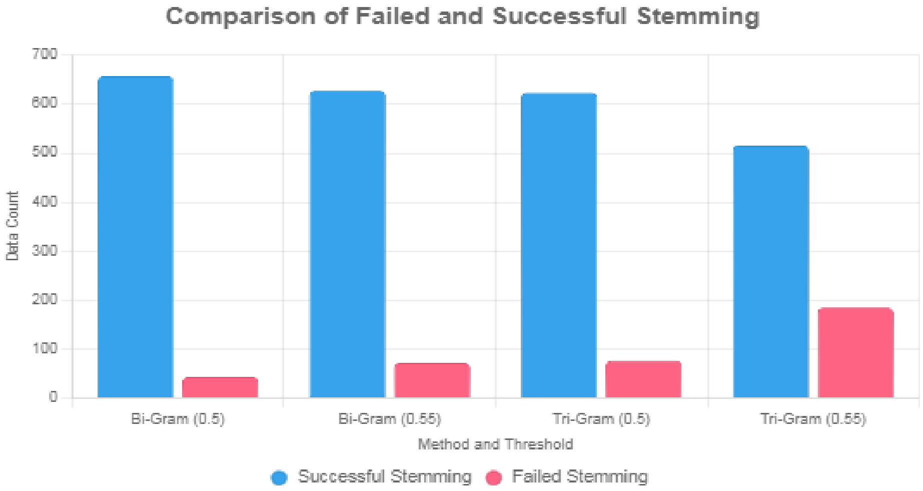

Fig 4. Comparison of Failed and Successful Stemming

This result indicates that the morphological structure of the dataset is more compatible with the bi-gram representation, which captures the pattern of two consecutive characters, whereas the tri-gram representation suffers from significant sparsity issues. Bi-gram stemming with a threshold of 0.5 proved to be the optimal configuration, providing the best balance of high accuracy, low failure rate, and consistent performance stability for the stemming system implementation.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

A. Experimental Findings on 700 Test Words with Bi-gram Stemming

In the Bi-Gram Stemming method with a threshold of 0.5, the test results show that out of a total of 700 words tested, 43 words fail to stem (Failed Stemming), while 657 words are successfully stemmed (Successful Stemming). Thus, the accuracy of this method reaches 0.9386 or 93.86%. Table 2 shows that the Bi-Gram Stemming method, with a threshold of 0.5, has a fairly high success rate in identifying the basic form of words in *nyo* dialects. The stemming performance decreases slightly when the threshold is increased to 0.55 in the Bi-Gram Stemming method. Out of 700 words tested, 72 words failed to stem, while 628 were successfully stemmed. The accuracy of this method is 0.8971 or 89.71%, which is lower than the threshold of 0.5. This shows that increasing the threshold in Bi-Gram Stemming tends to increase the number of stem failures, which may be due to stricter criteria for matching words.

Table 2. Bi-gram Stemming Result in 700 Data Test

Test Word	**Bi-Gram Stemming**	
	Threshold 0.5	**Threshold 0.55**
Successful Stemming Words	657	628
Failed Stemming Words	43	72
Total Words	700	700
Accuracy	0.9386	0.8971

B. Detailed Explanation of Bi-Gram Stemming Results

The description related to the results of Bi-gram stemming needs to consider words that are successfully stemmed and words that fail to be stemmed, both at a threshold of 0.5 and 0.55. The description uses tables and is represented by five samples each in the section (a) successful stemming with a threshold of 0.5, (b) successful stemming with a threshold of 0.55, (c) failed stemming with a threshold of 0.5, and (d) failed stemming with a threshold of 0.55.

Table 3. Sample of Successful Stemming with a Threshold of 0.5

No	**Test Word**	**Stemming Result**	**Base Word**	**Similarity Score**	**N-Gram Size**	**Threshold**
1	*adikmeu*	*adik*	*adik*	0.666666667	2	0.5
2	*akuklah*	*akuk*	*akuk*	0.666666667	2	0.5
3	*anaknou*	*anak*	*anak*	0.666666667	2	0.5
4	*andepei*	*andep*	*andep*	0.8	2	0.5
5	*anduknou*	*anduk*	*anduk*	0.727272727	2	0.5

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

The main information from tables 3 and 4 is that the stemming result column and base word column show the same word, and the similarity score value is greater than the threshold value.

Table 4. Sample of Successful Stemming with a Threshold of 0.55

No	Test Word	Stemming Result	Base Word	Similarity Score	N-Gram Size	Threshold
1	*pepohan*	*pepoh*	*pepoh*	0.8	2	0.55
2	*perebutken*	*rebut*	*rebut*	0.615384615	2	0.55
3	*pererok*	*rerok*	*rerok*	0.888888889	2	0.55
4	*peritungken*	*itung*	*itung*	0.571428571	2	0.55
5	*podaknou*	*podak*	*podak*	0.727272727	2	0.55

Some facts found in the 5 and 6 tables are: (1) stemming result and base word columns show different words, (2) if the similarity score column is zero, then the stemming result is the same as the test word, (3) even though the similarity score value is greater than the threshold value related to the stemming result column, an exception gives different results from the base word.

Table 5. Sample of Failed Stemming with a Threshold of 0.5

No	Test Word	Stemming Result	Base Word	Similarity Score	N-Gram Size	Threshold
1	*acak-acakan*	*cakak*	*acak*	0.6	2	0.5
2	*dibouken*	*dibouken*	*bou*	0	2	0.5
3	*dicetken*	*dicetken*	*cet*	0	2	0.5
4	*dijengei*	*dengei*	*jeng*	0.666666667	2	0.5
5	*disaiken*	*kedis*	*sai*	0.545454545	2	0.5

Table 6. Sample of Failed Stemming with a Threshold of 0.55

No	Test Word	Stemming Result	Base Word	Similarity Score	N-Gram Size	Threshold
1	*diwawaiken*	*diwawaiken*	*wawai*	0	2	0.55
2	*jengei*	*dengei*	*jeng*	0.8	2	0.55
3	*kanei*	*tanei*	*kan*	0.75	2	0.55
4	*maculken*	*maculken*	*pacul*	0	2	0.55
5	*majak*	*jajak*	*pajak*	0.857142857	2	0.55

C. Experimental Findings on 700 Test Words with Tri-gram Stemming

The Tri-Gram Stemming method with a threshold of 0.5 shows lower performance than Bi-Gram Stemming. Out of a total of 700 words, 76 words failed to be stemmed, and 624 words were successfully stemmed. The accuracy of this method is 0.8914 or 89.14%. Although the accuracy is still quite good, this method shows a higher failure rate than Bi-Gram Stemming at the same

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

threshold, which may be due to the Tri-Gram approach, which is more complex in grouping words. When the threshold of the Tri-Gram Stemming method is increased to 0.55, the stemming performance decreases. Of the 700 words tested, 184 failed to be stemmed, while 516 were successfully stemmed. The accuracy of this method is 0.7371 or 73.71%, the lowest accuracy among all tested methods. Increasing the threshold in Tri-Gram Stemming seems to significantly impact the failure rate, indicating that this method is more sensitive to threshold changes than Bi-Gram Stemming.

Table 7. Tri-gram Stemming Result in 700 Data Test

Test Word	**Tri-Gram Stemming**	
	Threshold 0.5	**Threshold 0.55**
Successful Stemming Words	624	516
Failed Stemming Words	76	184
Total Words	700	700
Accuracy	0.8914	0.7371

D. Detailed Explanation of Tri-Gram Stemming Results

The description connected to the results of Tri-gram stemming needs to incorporate words that are successfully stemmed and words that fail to be stemmed, both at a threshold of 0.5 and 0.55. The description employs tables and is represented by five examples each in the section (a) successful stemming with a threshold of 0.5, (b) successful stemming with a threshold of 0.55, (c) unsuccessful stemming with a threshold of 0.5, and (d) unsuccessful stemming with a threshold of 0.55. Table 8 provides the results of a stemming test for a sample of five words in the *Nyo* dialect using Tri-Gram stemming with a threshold of 0.5. Each test case provides the test word, stemming result, base word, similarity score, N-Gram size, and threshold. The test words—*behambughan*, *berhasil*, *bejangguk*, *bejawohan*, and *bekaccing-kaccingan*—yielded stemming results that matched their respective base words (*hambugh*, *hasil*, *jangguk*, *jawoh*, and *kaccing*), with similarity scores ranging from 0.588235294 (for *bekaccing-kaccingan*) to 0.833333333 (for *bejangguk*). All examples used a Tri-Gram method (N-Gram size of 3) and a threshold of 0.5, suggesting consistent and successful stemming performance across the sample.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

Table 8. Sample of Successful Stemming with a Threshold of 0.5

No	Test Word	Stemming Result	Base Word	Similarity Score	N-Gram Size	Threshold
1	*behambughan*	*hambugh*	*hambugh*	0.714285714	3	0.5
2	*behasil*	*hasil*	*hasil*	0.75	3	0.5
3	*bejangguk*	*jangguk*	*jangguk*	0.833333333	3	0.5
4	*bejawohan*	*jawoh*	*jawoh*	0.6	3	0.5
5	*bekaccing-kaccingan*	*kaccing*	*kaccing*	0.588235294	3	0.5

Table 9 shows findings from a stemming test for five words in the *Nyo* dialect using Tri-Gram stemming with a threshold of 0.55. It has columns for test number, test word, stemming result, base word, similarity score, N-Gram size, and threshold. The test words—*dipacul*, *dipaculei*, *dipaghokei*, *dipakkulei*, and *dipanasei*—produced stemming results (*pacul, pacul, paghok, pakkul,* and *panas*) that matched their respective base words, with similarity scores ranging from 0.6 (for *dipaculei* and *dipanasei*) to 0.75 (for *dipacul*). All cases utilised a Tri-Gram method (N-Gram size of 3) and a threshold of 0.55, indicating consistent and successful stemming performance across the dataset. The essential information from tables 8 and 9 is that the stemming result and base word columns indicate the same word, and the similarity score value is more than the threshold value.

Table 9. Sample of Successful Stemming with a Threshold of 0.55

No	Test Word	Stemming Result	Base Word	Similarity Score	N-Gram Size	Threshold
1	*dipacul*	*pacul*	*pacul*	0.75	3	0.55
2	*dipaculei*	*pacul*	*pacul*	0.6	3	0.55
3	*dipaghokei*	*paghok*	*paghok*	0.666666667	3	0.55
4	*dipakkulei*	*pakkul*	*pakkul*	0.666666667	3	0.55
5	*dipanasei*	*panas*	*panas*	0.6	3	0.55

Table 10. Sample of Unsuccessful Stemming with a Threshold of 0.5

No	Test Word	Stemming Result	Base Word	Similarity Score	N-Gram Size	Threshold
1	*naghei*	*aghei*	*taghei*	0.857142857	3	0.5
2	*ngajei*	*gergajei*	*kajei*	0.6	3	0.5
3	*ngebo*	*ngebo*	*bo*	0	3	0.5
4	*ngecahken*	*ngecahken*	*kecah*	0	3	0.5
5	*ngejuk*	*ngejuk*	*juk*	0	3	0.5

Some facts contained in the 10 and 11 tables are: (1) Stemming Result and Base Word columns show different words, (2) If the Similarity Score column is zero, then the Stemming Result is the same as the Test Word, (3) Even though the Similarity Score value is greater than the threshold value related to the Stemming Result column, an exception gives different results from the base word.

Table 11. Sample of Unsuccessful Stemming with a Threshold of 0.55

No	Test Word	Stemming Result	Base Word	Similarity Score	N-Gram Size	Threshold
1	*ngotou*	*botou*	*kotou*	0.571428571	3	0.55
2	*ngughusnou*	*ngughusnou*	*ughus*	0	3	0.55
3	*ngupei*	*upei*	*kupei*	0.666666667	3	0.55
4	*ngusikken*	*ngusikken*	*usik*	0	3	0.55
5	*nyapang*	*papang*	*capang*	0.666666667	3	0.55

Overall, Bi-Gram Stemming with a threshold of 0.5 showed the best performance with 93.85% accuracy, followed by Bi-Gram Stemming with a threshold of 0.55 (89.71%), Tri-Gram Stemming with a threshold of 0.5 (89.14%), and Tri-Gram Stemming with threshold 0.55 (73.71%). Bi-Gram Stemming is consistently superior to Tri-Gram Stemming at both threshold values, which suggests that the Bi-Gram approach may be more suitable for coastal dialects. In addition, increasing the threshold from 0.5 to 0.55 in both methods decreased accuracy, with a more significant decrease in Tri-Gram Stemming (89.14% to 73.71%) than in Bi-Gram Stemming (93.85% to 89.71%). These results show that choosing the right stemming method and threshold value is crucial in natural language processing, especially for coastal dialects. Bi-gram stemming with a threshold of 0.5 can be the best choice for applications requiring high stemming accuracy. However, adjustments to the threshold and method must be considered further for some instances that may require a stricter approach.

The failure of tri-gram stemming to achieve optimal performance is due to the fundamental data sparsity problem, where tri-grams do not appear frequently or at all in the training data, resulting in low probabilities and a very sparse feature space, with most combinations having insufficient representation in the dataset. The computational complexity of tri-grams increases exponentially compared to bi-grams, as the number of possible combinations of three consecutive characters is significantly larger. From a morphological perspective, tri-grams are too specific to capture inflection and derivation patterns in the language, especially in regional dialects such as the *nyo* dialect.

IV. CONCLUSION

Bi-Gram Stemming with a threshold of 0.5 emerges as the most effective method for stemming, based on the provided data, achieving an accuracy of 93.86% compared to 89.14% for

Tri-Gram Stemming at the same threshold. The superiority of the bi-gram approach is further demonstrated by its consistent performance across both test datasets while maintaining computational efficiency. When the threshold was increased to 0.55, performance dropped significantly to 89.71% for the bi-gram method and 73.71% for the tri-gram method, confirming that a threshold of 0.5 provides the optimal balance between precision and recall for the Lampung dialect of *Nyo* stemming. The underperformance of Tri-Gram Stemming, despite its more detailed character analysis, suggests that the complexity of triplet-based rules may introduce noise rather than improve accuracy for this particular dialect's morphological structure. This finding indicates that simpler n-gram approaches may be more suitable for languages with relatively straightforward affixation patterns. However, this study has several limitations. First, the evaluation was conducted on a relatively small dataset of 700 words, which may not fully represent the morphological diversity of the Lampung dialect of *Nyo*. Second, the research focused solely on accuracy metrics without considering computational efficiency or processing speed, which are crucial for real-world applications. Third, the stemming performance was not directly validated through downstream translation tasks, limiting our understanding of its practical impact on translation quality.

Future work should focus on integrating this stemming system into a comprehensive Lampung-to-Indonesian translation pipeline to validate its effectiveness in improving BLEU scores beyond the current 40% threshold. Additionally, expanding the evaluation dataset to include more diverse text sources and conducting comparative studies with other stemming algorithms would strengthen the findings. Investigation into hybrid approaches that combine the strengths of both bi-gram and tri-gram methods could also yield improved performance for specific word categories.

Author Contributions: *Parjito*: Conceptualization, Writing - Original Draft. *Zaenal Abidin*: Conceptualization, Methodology, Writing - Original Draft, Writing - Review & Editing Software, Investigation, Data Curation. *Akmal Junaidi*: Investigation, Data Curation, Supervision. *Wamiliana*: Formal Analysis, Methodology, Supervision. *Favorisen R. Lumbanraja*: Methodology, Supervision. *Farida Ariyani* : Resources, Validation.

All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Universitas Teknokrat Indonesia under HIPUTS Grand number 016/UTI/LPPM/E.1.1/IV/2023.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: The data and source code are available at the link attached to this paper.

Informed Consent: There were no human subjects.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

Animal Subjects: There were no animal subjects.

ORCID:
Parjito: https://orcid.org/0009-0006-9790-6459
Zaenal Abidin: https://orcid.org/0000-0003-4237-7167
Akmal Junaidi: https://orcid.org/0000-0003-1030-6954
Wamiliana: https://orcid.org/0000-0002-3740-7950
Favorisen R. Lumbanraja: https://orcid.org/0000-0002-1790-831X
Farida Ariyani: https://orcid.org/0000-0003-0937-0043

REFERENCES

[1] W. Hermawan, N. Eko, N. Udin, W. Akhyar, and E. Sanusi. “Sistem Morfologi Verba Bahasa Lampung Dialek Tulang Bawang”, Jakarta, Indonesia: Pusat Pembinaan dan Pengembangan Bahasa, Departemen Pendidikan Nasional, 2001.

[2] F. Ariyani, N. E. Rusminto, Sumarti, A. R. Idris, and L. Misliani, “Examining the Forms and Variations of the Lampung Script in Ancient Manuscripts,” *WSEAS Trans. Environ. Dev.*, vol. 18, pp. 204–217, 2022, doi: 10.37394/232015.2022.18.22.

[3] Z. Abidin, A. Junaidi, and Wamiliana, “Text Stemming and Lemmatization of Regional Languages in Indonesia: A Systematic Literature Review,” *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 2, pp. 217–231, Jun. 2024, doi: 10.20473/jisebi.10.2.217-231.

[4] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: state of the art, current trends and challenges,” *Multimed. Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: 10.1007/s11042-022-13428-4.

[5] J. Singh and V. Gupta, “A systematic review of text stemming techniques,” *Artif. Intell. Rev.*, vol. 48, no. 2, pp. 157–217, Aug. 2017, doi: 10.1007/s10462-016-9498-2.

[6] J. Singh and V. Gupta, “Text stemming: Approaches, applications, and challenges,” *ACM Comput. Surv.*, vol. 49, no. 3, Sep. 2016, doi: 10.1145/2975608.

[7] A. Jabbar, S. Iqbal, M. I. Tamimy, S. Hussain, and A. Akhunzada, “Empirical evaluation and study of text stemming algorithms,” *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5559–5588, Dec. 2020, doi: 10.1007/s10462-020-09828-3.

[8] J. Asian, H. E. Williams, and S. M. M. Tahaghoghi, “Stemming Indonesian,” in *Conferences in Research and Practice in Information Technology Series*, 2005, pp. 307–314. doi: 10.1145/1316457.1316459.

[9] A. Z. Arifin, H. T. Ciptaningtyas, P. Adhi, and K. Mahendra, "Enhanced confix stripping stemmer and ants algorithm for classifying news document in indonesian language." In *The International Conference on Information & Communication Technology and Systems*, vol. 5, pp. 149-158. 2009.

[10] F. Amin, W. Hadikurniawati, S. Wibisono, H. Februariyanti, and J. S. Wibowo, “A Hybrid Method of Rule-based and String Matching Stemmer for Javanese Language” *J. Theor. Appl. Inf. Technol.*, vol. 15, p. 19, 2017.

[11] M. A. Nq, L. P. Manik, and D. Widiyatmoko, “Stemming Javanese: Another Adaptation of the Nazief-Adriani Algorithm,” in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 627–631. doi: 10.1109/ISRITI51436.2020.9315420.

[12] S. I. Melia, J. Sholihah, D. Nisak, I. S. Juniaristha, and A. T. Ni’mah, “The Ngoko

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v10i1.25364

Javanese Stemmer uses the Enhanced Confix Stripping Stemmer Method," *Rekayasa*, vol. 16, no. 1, pp. 107–112, Apr. 2023, doi: 10.21107/rekayasa.v16i1.19308.

[13] N. W. Wardani and P. G. S. C. Nugraha, "Stemming Teks Bahasa Bali dengan Algoritma Enhanced Confix Stripping," *Int. J. Nat. Sci. Eng.*, vol. 4, no. 3, pp. 103–113, Dec. 2020, doi: 10.23887/ijnse.v4i3.30309.

[14] M. Agus, P. Subali, and C. Fatichah, "Kombinasi Metode Rule-based and N-Gram Stemming untuk Mengenali Stemmer Bahasa Bali," vol. 6, no. 2, pp. 219–228, 2019, doi: 10.25126/jtiik.201961105.

[15] J. Elektronik, I. K. Udayana, I. Gede, A. P. Arimbawa, N. Agus, and S. Er, "Lemmatization in Balinese Language". *Jurnal Elektronik Ilmu Komputer Udayana p-ISSN* 2301: 5373, 2017, doi: 10.24843/JLK.2020.v08.i03.p04.

[16] I.P.M. Wirayasa, I.M.A. Wirawan, and I.M.A. Pradnyana, "Algoritma Bastal: Adaptasi Algoritma Nazief & Adriani Untuk Stemming Teks Bahasa Bali," *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI*, *8*(1), pp.60-69, 2019.

[17] P. Gede Surya Cipta Nugraha and N. Wayan Wardani, "Stemming Dokumen Teks Bahasa Bali Dengan Metode Rule Base Approach," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 7, no. 3, pp. 510-521, 2020, doi: 10.35957/jatisi.v7i3.538.

[18] F. H. Rachman, N. Ifada, S. Wahyuni, G. D. Ramadani, and A. Pawitra, "ModifiedECS (mECS) Algorithm for Madurese-Indonesian Rule-Based Machine Translation," in *2022 International Conference of Science and Information Technology in Smart Administration, ICSINTESA 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 51–56. doi: 10.1109/ICSINTESA56431.2022.10041470.

[19] E. Lindrawati, E. Utami, and A. Yaqin, "Comparison of Modified Nazief&Adriani and Modified Enhanced Confix Stripping algorithms for Madurese Language Stemming," *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 7, no. 2, pp. 276–289, Aug. 2023, doi: 10.29407/intensif.v7i2.20103.

[20] Enni Lindrawati, Ema Utami, and A. Yaqin, "ANoM STEMMER: Nazief & Andriani Modification for Madurese Stemming," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 6, pp. 1341–1347, Dec. 2023, doi: 10.29207/resti.v7i6.5086.

[21] I. Setiawan and H. Y. Kao, "SUSTEM: An Improved Rule-based Sundanese Stemmer," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 23, no. 6, Jun. 2024, doi: 10.1145/3656342.

[22] A. Ardiyanti Suryani, D. Hendratmo Widyantoro, A. Purwarianti, and Y. Sudaryat, "The rule-based sundanese stemmer," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 17, no. 4, Jul. 2018, doi: 10.1145/3195634.

[23] A. Maesya, Y. Arifin, A. Zahra, and W. Budiharto, "Development of Sundanese Stemmer Based on Morphophonemics," in *10th International Conference on ICT for Smart Society, ICISS 2023 - Proceeding*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICISS59129.2023.10291840.

[24] A. Sutedi, R. Elsen, and M. R. Nasrulloh, "Sundanese Stemming using Syllable Pattern," *J. Online Inform.*, vol. 6, no. 2, p. 218, Dec. 2021, doi: 10.15575/join.v6i2.812.

[25] S. H. Wibowo and S. Wibowo, "Development of Stemming Algorithm for Rejang Language Stemmer Based on Rejang Language Morphology," *Artic. J. Adv. Res. Dyn. Control Syst.*, vol. 11, 2019.

[26] S. H. Wibowo, R. Toyib, M. Muntahanah, and Y. Darnita, "Time complexity in rejang language stemming," *J. INFOTEL*, vol. 14, no. 3, pp. 174–179, Aug. 2022, doi:

10.20895/infotel.v14i3.764.

[27] R. Sovia, S. Defit, Yuhandri, and Sulastri, "Development of natural language processing on morphology-based Minangkabau language stemming algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 31, no. 1, pp. 542–552, Jul. 2023, doi: 10.11591/ijeecs.v31.i1.pp542-552.

[28] R. Sovia, S. Defit, and Yuhandri, "Development of the Minangkabau Local Language Translation Machine Based on Stemming," in *Proceeding - 2022 International Symposium on Information Technology and Digital Innovation: Technology Innovation During Pandemic, ISITDI 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 195–198. doi: 10.1109/ISITDI55734.2022.9944457.

[29] Yusra, M. Fikry, and Hendi, "Stemmer bahasa melayu riau berdasarkan aturan morfologi." In *Seminar Nasional Teknologi Informasi Komunikasi dan Industri*, 2021, pp. 118-124.

[30] Z. Abidin, A. Wijaya, and D. Pasha, "Aplikasi Stemming Kata Bahasa Lampung Dialek Api Menggunakan Pendekatan Brute-Force dan Pemograman C#," *J. MEDIA Inform. BUDIDARMA*, vol. 5, no. 1, p. 1, Jan. 2021, doi: 10.30865/mib.v5i1.2483.

[31] Z. Abidin, A. Junaidi, Wamiliana, F. R. Lumbanraja, D. Kurniasari, and R. I. Borman, "Rule-Based Dialect of Tulang Bawang Stemmer," in *2025 International Conference on Advancement in Data Science, E-learning and Information System (ICADEIS)*, IEEE, Feb. 2025, pp. 1–6. doi: 10.1109/ICADEIS65852.2025.10933405.

[32] A. Guterres, Gunawan, and J. Santoso, "Stemming Bahasa Tetun Menggunakan Pendekatan Rule Based," *Teknika*, vol. 8, no. 2, pp. 142–147, Oct. 2019, doi: 10.34148/teknika.v8i2.224.

[33] A. Maesya, A. Ramadhan, E. Abdurachman, A. Trisetyarso, and M. Zarlis, "Stemming Algorithm for the Indonesian Language: A Scientometric View," in *2022 IEEE Creative Communication and Innovative Technology, ICCIT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICCIT55355.2022.10119050.

[34] Z. Abidin, P. Permata, and F. Ariyani, "Translation of the Lampung Language Text Dialect of Nyo into the Indonesian Language with DMT and SMT Approach," *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 5, no. 1, pp. 58–71, Feb. 2021, doi: 10.29407/intensif.v5i1.14670.