EVALUATING SIMILARITY MEASURES FOR MALAY NOISY TEXT NORMALIZATION: PERFORMANCE AND THRESHOLD ANALYSIS

Azilawati Azizan[1*] , Nurkhairizan Khairudin[2], Muhammad Fitri Shazwan Fadzely[3] , Nursyahidah Alias[4], Norshuhani Zamin[5], Norlina Mohd Sabri[6], Rohana Ismail[7]

[1, 2, 4] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Perak Branch, Tapah Campus, Perak, Malaysia

[3] Pembangunan Sumber Manusia Berhad, Wisma HRD Corp, 50490 Kuala Lumpur, Malaysia

[5] College of Computer Studies, De La Salle University, 2401 Taft Avenue, Manila, Philippines

[6] College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Terengganu Branch, Kuala Terengganu Campus, Terengganu, Malaysia

[7] Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut Campus, Terengganu, Malaysia

Emails: azila899@uitm.edu.my[1*], nurkh098@uitm.edu.my[2], fitrishazwan.work@gmail.com[3], syahidah@uitm.edu.my[4], norshuhani.zamin@dlsu.edu.ph[5], rohana@unisza.edu.my[7]

ABSTRACT

Noisy text normalization is a critical preprocessing step in natural language processing (NLP), particularly for user-generated content (UGC) that contains a lot of slang, abbreviations, and typographical errors. This extended study investigates the performance of multiple similarity measures in normalizing Malay noisy text, addressing gaps in prior study that predominantly relied on rule-based approaches and single similarity measures. By systematically evaluating token-based, edit-based, and sequence-based similarity measures across various thresholds, this study provides a comprehensive analysis of their effectiveness and computational efficiency. The methodology comprises a two-phase experiment: an initial phase to identify optimal thresholds using a small dataset and a second phase that generalizes findings on a larger dataset. Key findings reveal that edit-based measures, such as Levenshtein Distance and Damerau-Levenshtein, consistently outperform other measures at lower thresholds, achieving normalization success rates exceeding 83%. Ratcliff/Obershelp emerged as the most effective sequence-based measure, while token-based measures like Jaccard and Cosine demonstrated limited performance. The study also highlights the critical role of threshold in balancing normalization accuracy and flexibility. Additionally, computational time analysis underscores the trade-offs between accuracy and efficiency across similarity categories. These findings pave the way for more robust and adaptable text normalization strategies, particularly for Malay language studies.

Keywords: Noisy Text; Text Normalization; Malay Noisy Text; Similarity Measure; Threshold.

1.0 INTRODUCTION

The rapid growth of user-generated content (UGC) in social media, forums, and other online platforms has introduced significant challenges in natural language processing (NLP). One prominent challenge is the normalization of noisy text, which is essential for improving the performance of downstream NLP tasks such as sentiment analysis, machine translation, and text classification [1]. In the context of the Malay language, noisy text normalization remains underexplored, particularly in addressing issues such as informal spelling, slang, abbreviations, mixed language, and typographical errors frequently found in UGC [2].

Figure 1 shows is a single Malay sentence that incorporates multiple types of noisy text—slang, non-standard abbreviations, phonetic variations, spelling errors, regional dialect, and code-switching (mixed-language). That single sentence shows how diverse noise types can coexist in a single Malay sentence, posing challenges for text normalization.

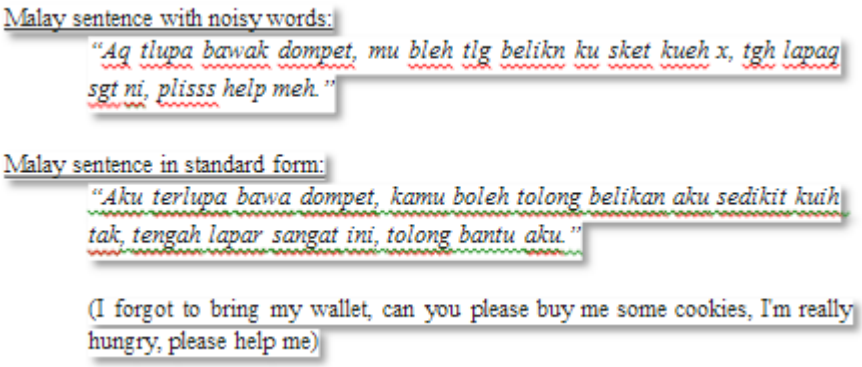

Fig. 1: Example of noisy words in a Malay sentence

Noisy text normalization aims to transform informal or non-standard text into its standard linguistic form. This process is particularly important for morphologically rich languages like Malay [3], where the structure and meaning of words are heavily influenced by prefixes, suffixes, and root words. The Malay language is considered morphologically rich, characterized by complex word formation processes such as affixation, reduplication, and compounding. A study titled "The Morphology of Malay" provides an in-depth analysis of these morphological processes, highlighting the language's complexity [4].

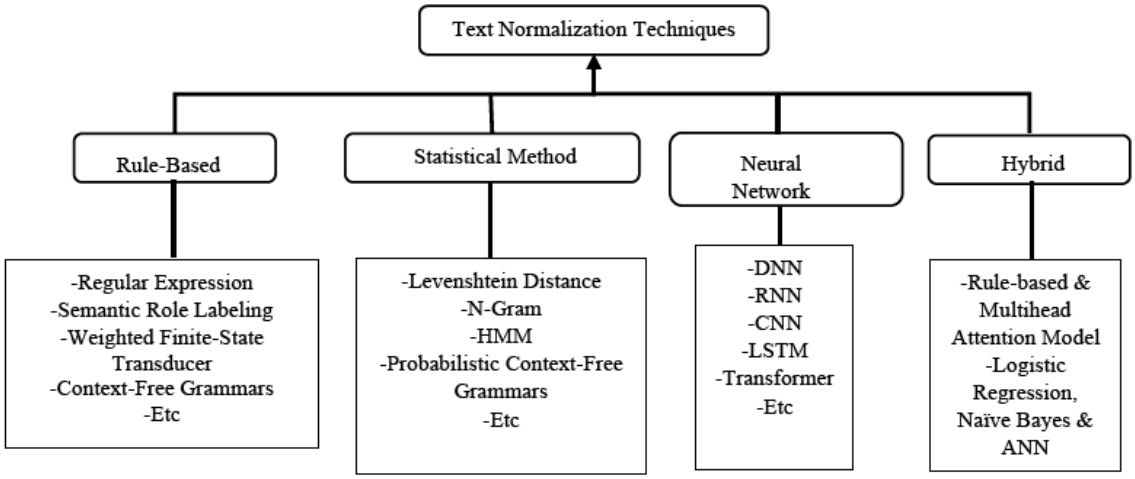

Fig. 2: Text normalization approaches [5]

Figure 2 depicts four major types of techniques used to normalise text. Traditionally, rule-based methods have been employed, relying on manually crafted transformation rules to handle specific patterns in noisy text. While effective for well-defined datasets, these approaches require substantial effort to create and maintain rules [6] as noisy text continuously evolves, necessitating frequent updates to address new patterns [7].

Besides rule-based methods, statistical approach referring to similarity measures represent another popular approach for normalizing noisy text [8]. Various types of similarity measures have been utilized, with Levenshtein Distance (LD) being the most common [9]. For example, LD calculates the minimum number of edits required to transform one word into another, making it effective for correcting typographical errors such as "teh" to "the" or "exmaple" to "example."

The most recent approach involves leveraging AI models to normalize noisy text, such as using transformer-based models like BERT and GPT [10] . For instance, BERT can be fine-tuned to handle Malay noisy text by training on a dataset of informal-to-formal text pairs, enabling it to predict the standardized form of a given noisy word or phrase. Similarly, sequence-to-sequence models [11], such as those used in machine translation, can be adapted for noisy text normalization tasks by treating noisy text as the input sequence and its normalized form as the output sequence.

While the latest advancements leverage AI models for noisy text normalization, there remains significant potential for further exploration and refinement of text similarity-based approaches. Various studies have explored noisy text normalization, few have systematically evaluated a wide range of similarity measures across multiple thresholds. For example, previous study has predominantly focused on single measures, such as LD or Jaccard, without examining their relative performance under different parameter settings [8]. By benchmarking multiple similarity measures, including edit-based, token-based, and sequence-based methods, this study offers a unique perspective on their comparative strengths and weaknesses.

Additionally, most prior studies have focused on well-resourced languages like English [12][13], leaving a gap in understanding how these methods perform for underrepresented languages like Malay. This study bridges that gap by providing empirical data specific to Malay noisy text, addressing the challenges posed by its linguistic structure and informal usage in UGC.

This study investigates the performance of various similarity measures in normalizing Malay noisy text by evaluating token-based, edit-based, and sequence-based similarity measures. It provides a comprehensive analysis of their effectiveness and computational efficiency. while placing significant emphasis on the impact of different thresholds on each measure's performance in normalizing Malay noisy words. Furthermore, this study focuses on normalization at the word level, as this serves as a foundational task for broader normalization processes in natural language processing. The primary objective is to identify the most suitable similarity measures for normalizing noisy Malay text, using thresholds optimized through comprehensive testing.

This study marks three contributions. First, it benchmarks the performance of a wide range of similarity measures, including LD, Damerau-Levenshtein, Jaccard, Cosine, and Ratcliff/Obershelp, among others, in the context of Malay noisy text normalization. Next, it identifies the optimal threshold for each similarity measure, balancing the trade-off between accuracy and computational cost. Finally, it provides practical insights into the applicability of similarity measures for real-world Malay text normalization tasks.

The remainder of this paper is organized as follows: Section 2 discusses related work in noisy text normalization and similarity measures. Section 3 outlines the methodology, including dataset preparation and evaluation criteria. Section 4 presents the experimental results and analysis. Section 5 concludes the study with key findings, limitation and future directions.

2.0 RELATED WORKS

This section reviews the evolution of noisy text normalization approaches, focusing on rule-based methods, similarity measures, and AI-driven techniques. Traditionally, noisy text normalization has been addressed using rule-based methods, which involve manually crafting transformation rules to handle common noise patterns. For example, studies by [14] demonstrated the efficacy of rule-based systems in handling typographical errors, abbreviations, and informal spelling. These systems rely heavily on predefined rules and lookup dictionaries, such as mapping “2day” to “today” or “gud” to “good” or “tp” to “tapi” (but), “x” to “tidak” (no). While effective for small datasets or specific domains, rule-based approaches are inherently limited by their reliance on static rules, making them less adaptable to evolving language patterns in UGC.

Text similarity measures offer a robust alternative to rule-based methods. It can be categorized into three categories; Edit-Based, Token-Based and Sequence-Based measures. These measures compute the degree of similarity between noisy and standard text, often serving as a foundation for identifying appropriate word replacements.

Edit-Based Similarity Measures, such as LD and Damerau-Levenshtein Distance, have been widely adopted for their ability to handle typographical errors. For instance, LD calculates the minimum number of character edits (insertions, deletions, substitutions) needed to transform one word into another [15]. Damerau-Levenshtein Distance extends this by accounting for transpositions, making it particularly useful for correcting errors like “teah” to “teach.”

Token-Based Measures like Jaccard, Cosine, and Sorensen-Dice are commonly used for longer text spans or phrases [16]. These measures evaluate similarity based on shared token sets or vectorized representations, which makes them suitable for detecting similar text fragments.

Sequence-based measures, such as Needleman-Wunsch, Smith-Waterman, and Ratcliff/Obershelp, are more specialized for sequence alignment tasks, capturing both local and global similarities. Studies by [17] highlight the effectiveness of these measures in normalizing noisy text, particularly in multilingual contexts.

The emergence of AI models has revolutionized noisy text normalization. Transformer-based models, such as BERT and GPT, have been successfully applied to normalization tasks by leveraging their contextual understanding of language. For instance, [18] fine-tuned a BERT and GPT-2 to normalize Arabic Dialect, achieving state-of-the-art results. Sequence-to-sequence models, often used in machine translation, have also been adapted for noisy text normalization [11]. These models treat the noisy input as a source sequence and the normalized text as a target sequence, enabling high accuracy in handling diverse noise patterns.

In essence, despite advancements in noisy text normalization approach, research on Malay text normalization remains limited. It presents unique challenges due to its rich morphology and extensive use of informal words, slang, and abbreviations in UGC. Early studies, such as [7], explored rule-based approaches for Malay text normalization, focusing on spelling corrections and dictionary lookups. However, these methods struggled to adapt to the dynamic nature of noisy Malay text. Recent efforts, many have begun exploring similarity measures and AI-based techniques for noisy text normalization [19]. However, even though AI models seem promising for contextually normalizing noisy text, they require extensive training on large datasets, which can be challenging for low-resource [20] languages like Malay.

Based on the literature, there are limited studies perform comprehensive evaluations across multiple similarity measures and thresholds specifically in the context of noisy text normalization. These are essential to assess the robustness and effectiveness of the approaches. Without testing across various thresholds and similarity measures, the potential of these techniques remains underexplored, and their applicability in real-world noisy text normalization scenarios may be limited. This gap highlights the need for more detail research in Malay noisy text normalization, particularly in benchmarking existing methods and exploring novel approaches.

3.0 METHODOLOGY

The methodology for this study references the conventional pipeline for normalizing noisy text. Typically, it involves the following stages: data acquisition, preprocessing, candidate identification, similarity scoring, selection of the best match, contextual adjustment, postprocessing, and outputting the normalized text [21]. However, this study focuses primarily on the similarity scoring and selection of the best match phases, while deferring the contextual adjustment stage for future study.

The objective of this study is to evaluate and compare the performance of various similarity measures at different levels for normalizing Malay noisy text. Sample data from UGC was collected and tested across multiple threshold settings to identify the optimal configuration for each similarity measure. The experiment was conducted in two phases: (1) an initial test on a small dataset to determine the best threshold for each similarity measure, and (2) a larger-scale test using the selected thresholds.

3.1 Data Acquisition

A sample dataset was collected from UGC on the X platform. The dataset consisted of 30 posts containing 634 words, with 68% classified as noisy words. For the first phase of the experiment, a subset of 15 words, including 10 noisy words, was selected from this dataset. In the second phase of the experiment, the entire dataset was used. The composition of the dataset is summarized in Table 1

Table 1: Dataset composition

Category of Words	Percentage
Noisy words	68%
English words	15%
Colloquial particle	9%
Standard words	8%

3.2 Preprocessing: Data Cleaning

The dataset underwent pre-processing to prepare it for normalization. Key steps included:

- Case Folding: Converting all text to lowercase.
- Removing Irrelevant Content: Eliminating special characters, URLs, and other extraneous elements.
- Tokenization: Splitting the text into individual words or tokens for further analysis.

Notably, stopwords were not removed because the objective was to normalize all words, including potentially misspelled stopwords, to maintain contextual integrity for downstream text analysis.

3.3 Candidate Identification

A noisy word refers to a word that deviates from the standard. Normalizing a noisy word means correcting the deviated word to its standard form [22]. This process involves comparing each noisy word to a standard word, which is referred to the Malay lexicon. In this study, the lexicon was derived from the Malaya Toolkit, which incorporates resources from the Pusat Rujukan Persuratan Melayu (PRPM) Dewan Bahasa dan Pustaka [23]. The lexicon contained 24,421 standard Malay words.

3.4 Similarity Scoring

The following similarity measures were selected for evaluation based on their suitability for text normalization tasks:

- **Edit-Based Measures**: Levenshtein Distance (LD), Damerau-Levenshtein, Jaro, and Jaro-Winkler.
- **Token-Based Measures:** Jaccard, Cosine Similarity, Sorensen-Dice, and Bag-of-Words (BoW).
- **Sequence-Based Measures:** Longest Common Subsequence (LCS), Needleman-Wunsch, Smith-Waterman, Gotoh, and Ratcliff/Obershelp.

Figure 3 illustrate the process of similarity scoring in the normalization of noisy Malay text. The process begins with a word input. The first step checks whether the input word exists in the Malay lexicon. If the word is already present in the lexicon, it is retained as is and marked as not requiring normalization.

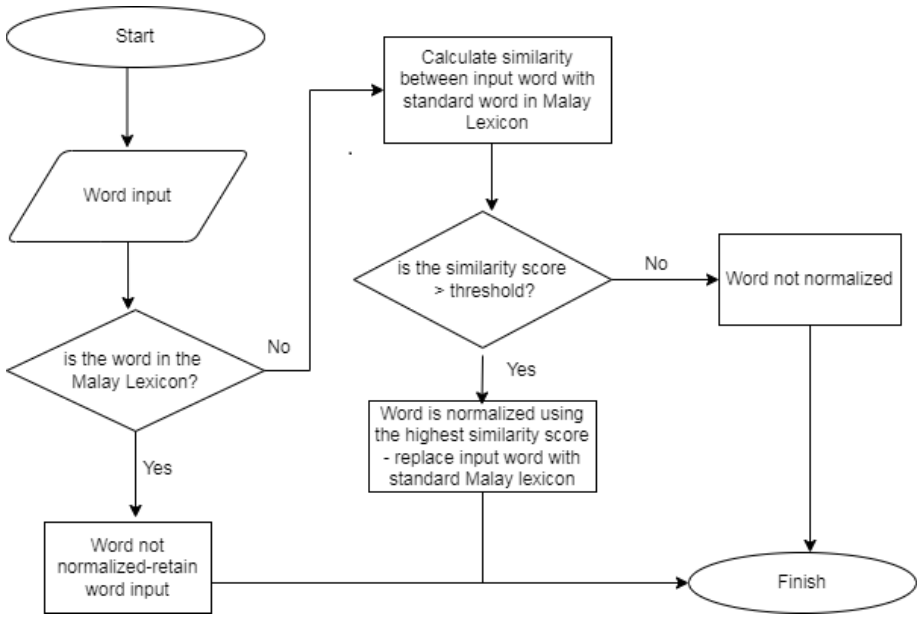

Fig. 3: Text normalization approaches

For words not found in the lexicon, a similarity scoring process is initiated. The input word is compared with standard words from the Malay lexicon to calculate similarity scores. If the highest similarity score exceeds a predefined threshold, the input word is replaced with the closest matching word from the lexicon, thus completing the normalization process. If no word in the lexicon meets the similarity score threshold, the word is marked as not normalized and retained in its original form.

The process end with either a normalized or retained word output. This flow ensures that only words with high similarity to standard forms are normalized while preserving unrecognizable or ambiguous words.

3.5 Selection of Best Match

Initial tests (first phase experiment) were conducted on a small subset of 15 words, of which 10 were noisy. A lexicon of 50 standard Malay words was used for similarity checking. The similarity between the noisy words and the lexicon was measured using all selected similarity measures, with thresholds ranging from 0.1 to 0.9. The best-

performing thresholds (optimal threshold) for each similarity measure were identified and subsequently applied in the second phase of the experiment.

In the second phase, the experiments were conducted on the larger dataset collected from UGC on the X platform (634 words). The noisy words were compared against a standard lexicon (24421 words) sourced from Dewan Bahasa dan Pustaka.

3.6 Output of Normalized Text

Normalized outputs were evaluated manually for accuracy. Successful normalization was defined as correcting noisy words while preserving the sequence of letters. For example, "sdap" normalized to "sedap" or "sadap" was deemed correct, while normalization to "pedas" or "asap" was considered incorrect.

4.0 RESULTS AND DISCUSSION

For these normalization experiments, Jupyter Notebook on Google Colab was used as the platform to execute all Python code. Several Python libraries are available for performing text similarity measurements, including TextDistance, RapidFuzz, FuzzyWuzzy, and Jellyfish [24]. We chose TextDistance as it offers a comprehensive collection of similarity modules, making it a versatile choice for our study.

The python codes for this experiment were constructed without incorporating any additional features. We utilized the original similarity measure calculations to establish a baseline performance for each similarity measure in normalizing Malay noisy text. This decision was made to evaluate the fundamental effectiveness of the similarity measures without the influence of enhancements.

4.1 First Phase Experiment Results

Figure 4 presents sample outputs from the normalization tests performed during this phase. Each similarity measure was evaluated across thresholds ranging from 0.1 to 0.9. The resulting output words were manually inspected and validated for correctness. Successfully normalized words were marked as 1, while unsuccessful attempts were recorded as 0

```
Levenshtein Normalized Results:
Threshold 0.1: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu di sekolah setiap hari sebab kawan suka belajar
Threshold 0.2: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu di sekolah setiap hari sebab kawan suka belajar
Threshold 0.3: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu di sekolah setiap hari sebab kawan suka belajar
Threshold 0.4: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu di sekolah setiap hari sebab kawan suka belajar
Threshold 0.5: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu di sekolah setiap hari sebab kawan suka belajar
Threshold 0.6: sy nak makin nasi itu dgn ayah di rumah shj tak mahu pi sekolah setiap hari sebab kawan suka belajar
Threshold 0.7: sy nk mkn nasi tu dgn ayah di rumah shj tak mahu pi sekolah setiap hari sbb kwn suka belajar
Threshold 0.8: sy nk mkn nasi tu dgn ayh di rumah shj tak mau pi sekolah setiap hari sbb kwn suka belajar
Threshold 0.9: sy nk mkn nasi tu dgn ayh di rumih shj tak mau pi sekolh stiap hari sbb kwn suka beljar

Jaccard Normalized Results:
Threshold 0.1: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu pergi sekolah setiap hari sebab kawan suka belajar
Threshold 0.2: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu pergi sekolah setiap hari sebab kawan suka belajar
Threshold 0.3: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu pergi sekolah setiap hari sebab kawan suka belajar
Threshold 0.4: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu pergi sekolah setiap hari sebab kawan suka belajar
Threshold 0.5: sayu nak makin nasi itu dengan ayah di rumah sahaja tak mahu pi sekolah setiap hari sebab kawan suka belajar
Threshold 0.6: sy nak makin nasi itu dgn ayah di rumah shj tak mahu pi sekolah setiap hari sebab kawan suka belajar
Threshold 0.7: sy nk mkn nasi tu dgn ayah di rumih shj tak mahu pi sekolah setiap hari sbb kwn suka belajar
Threshold 0.8: sy nk mkn nasi tu dgn ayh di rumih shj tak mau pi sekolah setiap hari sbb kwn suka belajar
Threshold 0.9: sy nk mkn nasi tu dgn ayh di rumih shj tak mau pi sekolh stiap hari sbb kwn suka beljar
```

Fig. 4: Sample of normalization output

The results were analysed, and the percentage of successfully normalized words was calculated. This study evaluates the percentage of successfully normalized words, which we refer to as Normalization Accuracy [25]. This metric is defined as the proportion of noisy words correctly normalized to their standard form. Formally, it is computed as:

$$Normalization\ Accuracy\ = \frac{Correctly\ Normalised\ Words}{Total\ Noisy\ Words} \times 100\%$$

This approach ensures a direct and interpretable assessment of each similarity measure's ability to normalize Malay noisy text. The terminology is adopted to align with prior text normalization research, which emphasizes word-level correction accuracy over classification-based precision and recall metrics [8], [14], [26] ,[27], [28]. Table 2

summarizes the findings. The results provide insights into the performance trends of different similarity measures, forming the basis for further analysis.

Table 2: Percentage of successful normalized word

Category	Similarity Measure	Threshold								
		0.1	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
Edit-Based Similarity Measures	Levensthein Distance (LD)	1	1	1	1	1	0.733	0.4	0.267	0
	Damerau-Levensthein	1	1	1	1	1	0.733	0.4	0.267	0
	Jaro	1	1	1	1	1	1	0.667	0.533	0.267
	Jaro-Winkler	1	1	1	1	1	1	0.667	0.6	0.4
Token-Based Similarity Measures	Jaccard	1	1	1	1	0.933	0.733	0.333	0.2	0
	Cosine	1	1	1	1	1	1	0.933	0.6	0.2
	Sorensen	1	1	1	1	1	0.933	0.733	0.533	0.2
	Bow	0.867	0.867	0.867	0.867	0.867	0.6	0.4	0.267	0
Sequence-based Similarity Measure	LCS	1	1	1	1	1	0.733	0.4	0.267	0
	Needleman-Wunsch	0.933	0.933	0.933	0.933	0.933	0.8	0.467	0.267	0.067
	Smith-Waterman	0.4	0.4	0.4	0.4	0.4	0.333	0.333	0.333	0.067
	Gotoh	0.933	0.933	0.933	0.933	0.933	0.867	0.6	0.4	0.267
	Ratcliff	1	1	1	1	1	0.933	0.733	0.533	0.2

LD, Damerau-Levenshtein, Sorensen, LCS, and Ratcliff: These measures achieved perfect normalization (100%) at lower thresholds (0.1 to 0.5). However, their performance dropped significantly as the threshold increased, becoming ineffective at thresholds ≥ 0.8. This suggests that these measures perform well with flexible (lower) thresholds but are more restrictive at higher thresholds.

Jaro and Jaro-Winkler: Both measures performed very well, maintaining 100% normalization for thresholds up to 0.6. However, Jaro-Winkler slightly outperformed Jaro at higher thresholds (e.g., 0.8).

Jaccard, Cosine: These measures began to diverge at thresholds around 0.5 and 0.7. Cosine performed slightly better than Jaccard, especially at thresholds 0.6–0.7, but both struggled significantly at higher thresholds.

Bag-of-Words (BoW): BoW consistently underperformed compared to other measures. It started with lower normalization percentages at even low thresholds (0.1–0.2) and declined sharply at thresholds ≥ 0.6.

Needleman-Wunsch and Gotoh: These measures were relatively stable at lower thresholds, maintaining around 93% accuracy up to 0.5. Gotoh slightly outperformed Needleman-Wunsch at higher thresholds.

Smith-Waterman: This measure had the lowest overall performance, peaking at only 40% even at low thresholds. It was ineffective for normalization at higher thresholds.

The key observations from these results can be discussed on three aspects. First, is the high sensitivity of thresholds. Threshold values have a significant impact on all similarity measures[5]. Lower thresholds (e.g., 0.1–0.5) allow for more flexible matches, which often leads to better normalization performance, especially for noisy text. Meanwhile, higher thresholds (≥0.8) tend to be too restrictive, causing many words to remain unnormalized or incorrectly matched.

Second observation is on the measure suitability for noisy text normalization. Best overall measures are LD, Damerau-Levenshtein, and Ratcliff, where they performed consistently well for thresholds ≤ 0.5. For balanced measures, Jaro-Winkler showed the best balance between flexibility and precision, maintaining good accuracy even at thresholds as high as 0.7–0.8. While for low suitability, Smith-Waterman and Bag-of-Words are less suited for this task due to their low performance across most thresholds.

The third observation is on the trade-off between flexibility and precision. Lower thresholds prioritize flexibility, allowing for normalization even with larger differences between noisy and lexicon words. Meanwhile, higher thresholds prioritize precision, which is useful when exact matches are needed but can fail for highly noisy text.

For a graphical comparison, Figure 5 visualize the the performance of various similarity measures across different thresholds. Each line represents a method, with the x-axis showing the threshold values and the y-axis showing the percentage of correctly normalized words.

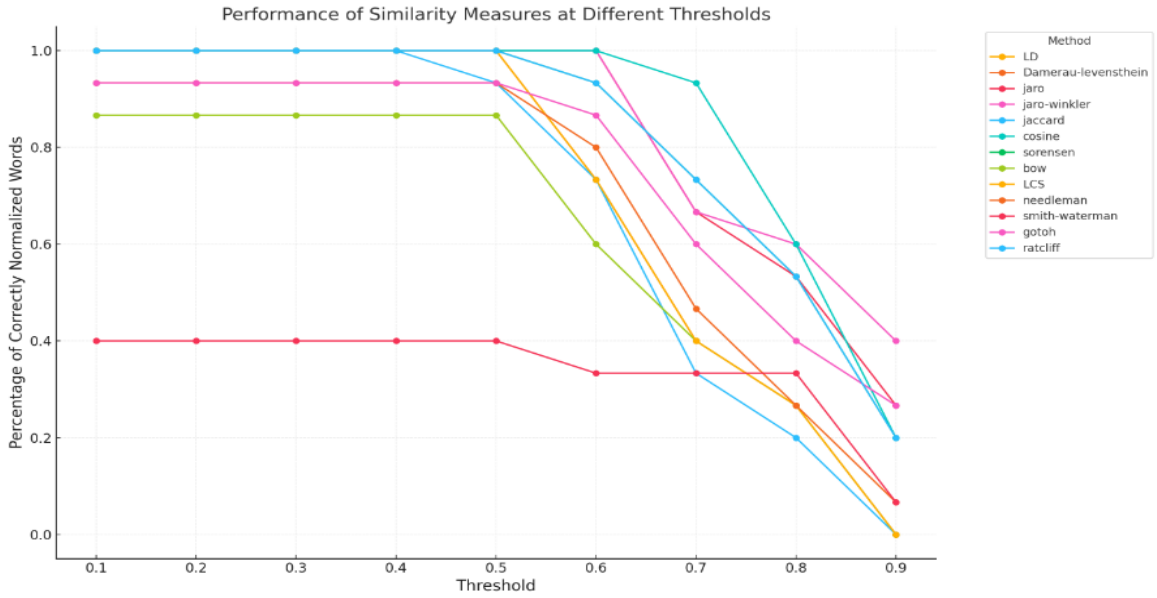

Fig. 5: Performance of similarity measures across different thresholds

The graph provides important insight such as, normalization perform higher at lower threshold (0.1–0.5), where most methods perform perfectly or near-perfectly (100% accuracy) at lower thresholds. This indicates their ability to match strings effectively when similarity thresholds are more lenient. Methods such as Levensthein Distance (LD), Damerau-Levenshtein, and Ratcliff consistently achieve 100% accuracy at lower thresholds.

The graph also shows a very significant divergence of performance at higher thresholds (0.6–0.9). The performance of almost all methods decreases as the threshold increases. Meanwhile Smith-Waterman exhibits the lowest performance overall, maintaining a consistent accuracy of 40% or less across all thresholds.

However, Jaccard and Cosine Similarity manage to maintain high performance up to a threshold of 0.6 but decline significantly after that. Meanwhile Needleman-Wunsch performs better than Smith-Waterman, particularly at thresholds between 0.1 and 0.6, and both performance dropped almost uniformly starting from a threshold of 0.7.

4.2 Second Phase Experiment Results

Table 3 presents the results, summarizing the percentage of successfully normalized words for each similarity measure. Unlike the first phase experiment (Table 2), which explores multiple threshold values for each similarity measure, second pahse experiment (Table 3) focuses solely on the best-performing threshold identified during small dataset testing. The purpose of the large dataset evaluation is not to re-examine all thresholds but to validate whether the optimal threshold generalizes effectively to a larger and more diverse dataset. Including all thresholds in Table 3 would not provide additional insights, as the first-phase experiment has already established which thresholds perform best for each similarity measure.

The results in the second phase experiment shows that in the category of edit-based measures, LD achieved the highest success rate among all measures with a 83.9% normalization accuracy at a threshold of 0.5, demonstrating its robustness for Malay noisy text normalization. Damerau-Levenshtein closely followed with a success rate of 83.0% at the same threshold. Its slight variation from LD in performance could be attributed to its ability to account for transpositions feature. Compared to prior studies, where LD has been used for Indonesian text normalization, achieving success rates ranging from 60% [8], 73% to 75% [29] and 78% to 79% [25], our findings demonstrate improved accuracy by optimizing the threshold at 0.5.

Meanwhile, Jaro and Jaro-Winkler performed moderately, achieving success rates of 67.9% and 66.1% respectively at thresholds of 0.6. These measures are typically suited for short strings but showed less effectiveness

compared to LD. However, our results indicate that Jaro-Winkler outperformed the findings in [8] where an accuracy of only 54.6% was reported.

On the other hand, token-based measures generally performed poorly compared to edit-based measures. Jaccard, Cosine, and Sorensen all demonstrated low success rates of 29.5%, suggesting their limited utility for normalizing Malay noisy text. These measures rely on set/token comparisons and may not be well-suited for noisy text with character-level variations. Bag-of-Words (BoW) had the lowest success rate among token-based measures at 26.8%, likely due to its simplistic representation of text, which does not capture structural or sequential information.

Table 3: Successfully normalized words

Category	Optimal Threshold	Similarity Measure	Normalization Accuracy
Edit-Based Similarity Measures	0.5	**Levenshtein Distance (LD)**	**0.839**
	0.5	**Damerau-Levenshtein**	**0.830**
	0.6	Jaro	0.679
	0.6	Jaro-Winkler	0.661
Token-Based Similarity Measures	0.4	Jaccard	0.295
	0.6	Cosine	0.295
	0.5	Sorensen	0.295
	0.5	BoW	0.268
Sequence-based Similarity Measure	0.5	LCS	0.679
	0.5	Needleman-Wunsch	0.580
	0.5	Smith-Waterman	0.259
	0.5	**Gotoh**	**0.723**
	0.5	**Ratcliff**	**0.786**

For the sequence-based similarity measures, Ratcliff/Obershelp showed strong performance with a success rate of 78.6% at a threshold of 0.5, making it the most effective sequence-based measure. Gotoh also performed well, achieving a success rate of 72.3% at the same threshold. Its incorporation of gap penalties likely contributed to its robustness in text normalization. Meanwhile, Longest Common Subsequence (LCS) achieved a moderate success rate of 67.9%, indicating its ability to match sequences effectively but with limitations compared to other measures. Needleman-Wunsch and Smith-Waterman displayed lower success rates of 58.0% and 25.9%, respectively. These measures, designed for sequence alignment in biological data, may not be as adaptable to noisy text normalization tasks.

To summarize the findings, edit-based similarity measures consistently outperformed token-based and sequence-based measures, with LD and Damerau-Levenshtein being the most effective overall. While, Ratcliff/Obershelp stood out among sequence-based measures as a strong contender, demonstrating competitive accuracy. Token-based similarity measures, including Jaccard, Cosine, and Sorensen, showed the lowest success rates, indicating their unsuitability for character-level noise and structure-specific text normalization. In addition, the thresholds were optimized for performance, demonstrating that fine-tuning the threshold for each measure significantly affects the normalization success rate. For instance, LD and Damerau-Levenshtein achieve peak performance at a threshold of 0.5, while Jaro and Jaro-Winkler perform best at 0.6 threshold.

Practical recommendations for Malay noisy text normalization:

- For Malay noisy text normalization, measures like **LD, Damerau-Levenshtein**, and **Ratcliff** should be preferred when working with lower thresholds (0.3–0.5), especially when dealing with highly noisy text.
- **Jaro-Winkler** offers a balanced approach to normalization, making it particularly effective when using higher thresholds (0.6–0.7) to ensure both flexibility and accuracy in matching noisy text.
- Measures like **Smith-Waterman** are not ideal for text normalization tasks and should likely be avoided.

In addition, the performance of similarity measures is significantly influenced by the characteristics of the text. For instance, short words or abbreviations tend to benefit more from flexible measures like Damerau-Levenshtein, while text with numerous transposed characters is better managed by Ratcliff or Jaro-Winkler. Among edit-based measures, LD stands out for its balance of accuracy and computational efficiency. These findings highlight the importance of selecting similarity measures that align with the specific characteristics of noisy datasets, such as Malay user-generated content.

Besides normalization accuracy, the time required for the normalization process was also recorded, as shown in Table 4. The comparison reveals insights into the computational efficiency of each similarity category.

Table 4: Normalization processing time

Similarity Category	**Time Taken to Normalize**
Edit-Based Similarity Measures (LD, Damerau-levenshtein, jaro&jaro-winkler)	1m 8s
Token-Based Similarity Measures (jaccard, cosine-similarity, sorensen-dice,bow)	2m 43s
Sequence-based Similarity Measures(LCS, Needleman-Wunsch, Smith-Waterman, Gotoh, Ratcliff/Obershelp)	2m 41s

Edit-based similarity measures were the fastest to compute, taking only 1 minute and 8 seconds on average, making them computationally efficient. Token-based and sequence-based measures required significantly longer processing times, around 2 minutes 40 seconds, due to the complexity of their underlying algorithms. These findings align with those reported in [8], which also showed that edit-based similarity measures normalized faster than other approaches. In conclusion, LD offers the best trade-off between accuracy and computational efficiency for normalizing Malay noisy text based on the results of this experiment.

4.3 Comparison of First and Second Phase Experiment Results

The performance evaluation of similarity measures across different dataset sizes reveals a notable trend of slight performance degradation when applied to the larger dataset. Table 5 exhibits the accuracy comparison between the first- and second-phase experiments. While the first-phase experiment (small dataset) identified the optimal thresholds for each measure, the second-phase experiment (large dataset) demonstrates that even the best-performing similarity measures experienced a decline in accuracy when tested on a more diverse and complex dataset.

Table 5: Accuracy comparison of first and second phase results

Category	**Optimal Threshold**	**Similarity Measure**	**Normalization Accuracy**	
			First Phase	**Second Phase**
Edit-Based Similarity Measures	0.5	Levenshtein Distance (LD)	1	0.839
	0.5	Damerau-Levenshtein	1	0.830
	0.6	Jaro	1	0.679
	0.6	Jaro-Winkler	1	0.661
Token-Based Similarity Measures	0.4	Jaccard	1	0.295
	0.6	Cosine	1	0.295
	0.5	Sorensen	1	0.295
	0.5	BoW	0.867	0.268
Sequence-based Similarity Measure	0.5	LCS	1	0.679
	0.5	Needleman-Wunsch	0.933	0.580
	0.5	Smith-Waterman	0.4	0.259
	0.5	Gotoh	0.933	0.723
	0.5	Ratcliff	1	0.786

Edit-Based Measures, such as Levenshtein Distance (LD) and Damerau-Levenshtein, performed exceptionally well in the small dataset, with near-perfect success rates. However, in the large dataset, their success rates dropped slightly, with LD declining from 100% in small-scale testing to 83.9% in large-scale testing. This indicates that while these methods are highly effective for localized word distortions, they struggle when handling a broader range of noisy variations present in larger datasets.

Jaro and Jaro-Winkler exhibited a moderate performance drop. Both measures showed moderate success in small-scale testing, performing well on short words and minor character swaps. However, when tested on a larger dataset, their performance declined further, suggesting that their reliance on prefix weighting is effective for limited vocabulary settings but may not generalize well to highly varied noisy text.

Meanwhile, Token-Based Measures (Jaccard, Cosine, Sorensen-Dice, BoW) performed moderately in the small dataset, but their performance dropped significantly in the larger dataset, never exceeding 30% accuracy. Their set-based approach is inherently less effective for word-level noisy text normalization, reinforcing their limited applicability in this context.

In the Sequence-Based Measures category, Ratcliff remained one of the most stable measures, maintaining high performance (78.6%) even in large dataset testing. Smith-Waterman, on the other hand, struggled significantly (25.9%), further emphasizing its limited suitability for noisy text normalization. Meanwhile, Gotoh demonstrated resilience, achieving 72.3% accuracy, making it one of the better sequence-based measures.

The consistent drop in success rates across all measures suggests that as the dataset size increases, noisy text variations become more complex and harder to normalize. While edit-based measures remain the most effective, they require further refinements to maintain high performance in larger, real-world datasets. Sequence-based methods like Ratcliff/Obershelp and Gotoh show promising results, but their efficiency needs improvement for scalability. Token-based measures consistently underperform, confirming that they are not well-suited for character-level noisy text normalization tasks.

These findings highlight that while similarity measures can effectively normalize Malay noisy text, their performance is affected by dataset scale and complexity.

5.0 CONCLUSION AND FUTURE WORK

This extended study provides a comprehensive analysis of similarity measures for normalizing Malay noisy text. By systematically evaluating edit-based, token-based, and sequence-based similarity measures across various thresholds, the study offers valuable insights in terms of performance, computational efficiency, and practical applicability.

Key findings demonstrate that edit-based measures, particularly LD and Damerau-Levenshtein, consistently outperform other methods at lower thresholds, achieving normalization success rates exceeding 83%. Among sequence-based measures, Ratcliff/Obershelp emerges as the most robust, while token-based measures like Jaccard and Cosine show limited utility for this task. The study also highlights the importance of threshold optimization in balancing accuracy and flexibility, a crucial factor for real-world noisy text normalization tasks.

While this study provides a comprehensive analysis of similarity measures and thresholds, it has certain limitations. Firstly, it relies on a static lexicon of standard Malay words, which may not capture emerging slang or newly coined terms in UGC. Secondly, the study does not incorporate contextual adjustment, which could improve normalization accuracy by leveraging sentence-level information. Finally, the computational efficiency of each measure, while discussed, could benefit from further optimization for large-scale applications.

Future work could address these limitations by integrating dynamic lexicons that adapt to evolving language patterns and exploring machine learning-based approaches for context-aware normalization. Additionally, combining multiple similarity measures to create hybrid methods could further enhance performance and balance the strengths of different approaches. The integration of AI-driven models, such as transformer-based architectures, also holds promise for improving context-aware normalization. Additionally, extending this study to other languages can further generalize the findings and contribute to the global NLP community.

In conclusion, noisy text normalization has evolved from rule-based approach to similarity measures and AI-driven techniques, each offering distinct advantages and limitations. Despite these advancements, significant gaps remain in the evaluation and refinement of these approaches for Malay text normalization.. These findings present the opportunity to develop more effective and adaptable normalization strategies, particularly for low-resource languages like Malay.

ACKNOWLEDGMENT

The authors would like to thank Universiti Teknologi MARA (UiTM), Perak branch, Tapah campus for providing the opportunity, support, and facilities necessary to successfully carry out this project. Special thanks are extended to the Multidisciplinary Information Retrieval (MuDIR) research interest group for their encouragement throughout this study

REFERENCES

[1] A. Upadhye, “A Comprehensive Survey of Text Data Cleaning Techniques : Challenges , Methods , and Best Practices,” *J. Sci. Eng. Res.*, vol. 7, no. 8, pp. 205–210, 2020.

[2] M. A. Saloot, N. Idris, and R. Mahmud, “An architecture for Malay Tweet normalization,” *Inf. Process. Manag.*, vol. 50, no. 5, pp. 621–633, 2014.

[3] H. Abdullah, “The Morphology of Malay,” University of Edinburgh, 1972.

[4] M. Maziyah, M. Melvin, J. Y. Qian, W. Chee, D. Jared, and D. Jared, “Malay Lexicon Project 2 : Morphology in Malay word recognition,” *Mem. Cognit.*, no. June 2022, pp. 647–665, 2023.

[5] A. A. Aliero and N. M. Dankolo, “Systematic Review on Text Normalization Techniques and its Approach to Non-Standard Words,” *Int. J. Comput. Appl.*, vol. 185, no. 33, pp. 44–55, 2023.

[6] Yang Zhang; Evelina Bakhturina ; Aleksandra Antonova, “Text Normalization and Inverse Text Normalization with NVIDIA NeMo,” *NVIDIA Technical Blog*, 2022. .

[7] S. N. A. N. Ariffin and S. Tiun, “Rule-based text normalization for malay social media texts,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, pp. 156–162, 2020.

[8] A. Yohni, W. Finansyah, and V. M. Sutanto, “Performance Comparison of Similarity Measure Algorithm as Data Preprocessing Stage : Text Normalization in Bahasa Indonesia,” *Sci. J. Informatics*, vol. 9, no. 1, pp. 1–7, 2022.

[9] A. Mehta, V. Salgond, D. Satra, and N. Sharma, “Spell Correction and Suggestion Using Levenshtein Distance,” *Int. Res. J. Eng. Technol.*, pp. 1977–1981, 2021.

[10] M. S. and Ü. E. Ölçer, “Impact of Transformer-Based Models in NLP: An In-Depth Study on BERT and GPT,” *8th Int. Artif. Intell. Data Process. Symp.*, 2024.

[11] I. Lourentzou, K. Manghnani, and C. X. Zhai, “Adapting sequence to sequence models for text normalization in social media,” *Proc. 13th Int. Conf. Web Soc. Media, ICWSM 2019*, no. Icwsm, pp. 335–345, 2019.

[12] E. Flint, E. Ford, O. Thomas, and A. Caines, “A Text Normalisation System for Non-Standard English Words,” in *3rd Workshop on Noisy User-generated Text*, 2017, pp. 107–115.

[13] Y. Ehara, “To What Extent Does Lexical Normalization Help English-as-a-Second Language Learners to Read Noisy English Texts ?,” 2018.

[14] M. Toska, “A Rule-Based Normalization System for Greek Noisy User-Generated Text,” Uppsala University, 2020.

[15] D. K. Po, “Similarity Based Information Retrieval Using Levenshtein Distance Algorithm,” *Int. J. Adv. Sci. Res. Eng.*, vol. 06, no. 04, pp. 06–10, 2020.

[16] R. . Verma, V., Aggarwal, “A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: empirical and theoretical perspective,” *Soc. Netw. Anal. Min.*, 2020.

[17] D. Kubal, “Multilingual Sequence Labeling Approach to solve Lexical Normalization,” pp. 457–464, 2021.

[18] Khalid Alnajjar; Mika Hämäläinen ;, “Normalization of Arabic Dialects into Modern Standard Arabic using BERT and GPT-2,” *J. Data Min. Digit. Humanit.*, pp. 1–8, 2024.

[19] J. Wang and Y. Dong, “Measurement of Text Similarity : A Survey,” *Information*, 2020.

[20] Y. Yeong, T. Tan, and S. K. Mohammad, “Using Dictionary and Lemmatizer to Improve Low Resource English-Malay Statistical Machine Translation System,” *Procedia - Procedia Comput. Sci.*, vol. 81, no. May, pp. 243–249, 2016.

[21] S. Kumar, “Text Normalization,” in *Python for Accounting and Finance*, Springer Nature, 2024.

[22] A. Azizan, N. H. Anuar, N. Khairuddin, and R. Ismail, “Normalization of Malay Noisy Text in Social Media using Levenshtein Distance and Rule-Based Techniques,” vol. VIII, no. 2454, pp. 1535–1544, 2024.

[23] “Pusat Rujukan Persuratan Melayu-Dewan Bahasa & Pustaka,” *Dewan Bahasa & Pustaka (DBP)*, 2017. [Online]. Available: https://prpm.dbp.gov.my/. [Accessed: 22-Sep-2024].

[24] W. Andrzejewski, B. Bębel, P. Boiński, M. Sienkiewicz, and R. Wrembel, “Text similarity measures in a data deduplication pipeline for customers records,” *CEUR Workshop Proc.*, vol. 3369, pp. 33–42, 2023.

[25] R. P. Kusumawardani, S. Priansya, and F. J. Atletiko, “Context-sensitive normalization of social media text in bahasa Indonesia based on neural word embeddings,” *Procedia Comput. Sci.*, vol. 144, pp. 105–117, 2018.

[26] O. Tursun, “Noisy Uyghur Text Normalization,” in *Proceedings ofthe 3rd Workshop on Noisy User-generated Text*, 2017, pp. 85–93.

[27] M. Bollmann, “A large-scale comparison of historical text normalization systems,” *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1, pp. 3885–3898, 2019.

[28] T. T. T. Nguyen, X. B. Dang, and X. T. Nguyen, “A hybrid method for Vietnamese text normalization,” *ACM Int. Conf. Proceeding Ser.*, no. ii, pp. 104–109, 2019.

[29] P. Santoso, P. Yuliawati, R. Shalahuddin, and A. P. Wibawa, “Damerau Levenshtein Distance for Indonesian Spelling Correction,” *J. Inform.*, vol. 13, no. 2, p. 11, 2019.