Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P); 2583-7907 (E)
Vol.14 No.2, 2025, pp.51-55

www.crijournals.org
DOI: https://doi.org/10.70112/ajcst-2025.14.2.4389

N-gram Based Lexical Sentence Similarity Score Using Modified Jaccard Algorithm

*P. Senthilkumar and K. Nandhini
Department of Computer Science, Central University of Tamil Nadu, Tamil Nadu, India
E-mail: nandhinikumaresh@cutn.ac.in
*Corresponding Author: way2sen@gmail.com
(Received 10 November 2025; Revised 25 November 2025; Accepted 28 November 2025; Available online 5 December 2025)

Abstract **- In recent years, the Internet has evolved into a global phenomenon, making it nearly impossible to envision modern life without it. Among the vast forms of online content, textual data holds the greatest significance due to its abundance and informational value. However, managing and analysing such extensive text corpora poses several challenges, with sentence similarity emerging as one of the most complex problems in Natural Language Processing (NLP). Although existing sentence comparison techniques perform effectively in specific contexts, they often struggle in others and typically require substantial computational resources, including powerful hardware, extensive training datasets, and high processing capabilities. To address these limitations, this study introduces a lightweight approach that emphasizes word-level similarity through comprehensive n-gram comparisons. The proposed method incorporates semantic understanding and evaluates the longest common subsequence within sentences to generate a more accurate similarity score. It demonstrates superior efficiency over baseline methods by minimizing computational requirements and leveraging straightforward mathematical operations.**

Keywords**: Natural Language Processing (NLP), Sentence Similarity, Answer Assessment, Jaccard Algorithm, Subjective Answer Evaluation**

I. INTRODUCTION

Sentence similarity is widely used in various real-time applications, such as content search, information retrieval [1], plagiarism detection, question answering, and the evaluation of subjective answers [2]. The similarity score can be determined using various methods, each with its own advantages and disadvantages. Some methods are better suited for short sentences [3], while others are more effective for longer ones. Certain approaches also require significant computing resources, large volumes of trained data, and human oversight [4]. The sentence similarity methods can be divided into three main categories, as illustrated in Figure 1: the lexical-based approach, the semantic-based approach, and the deep learning-based approach. The lexical approach measures similarity based on individual tokens in a sentence [5], while the semantic approach considers the overall meaning [1,5]. The deep learning-based approach incorporates both lexical and semantic elements.

$$\text{Cosine Similarity}(A, B) = (A \cdot B) / (\|A\|\|B\|) \quad (1)$$

$$\text{Jaccard Similarity}(A, B) = |A \cap B| / |A \cup B| \quad (2)$$

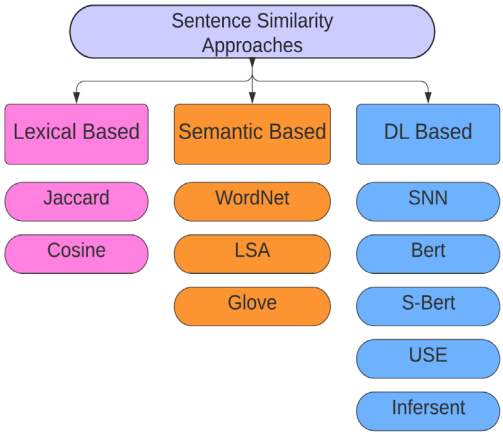

Fig.1 Classification of Sentence Similarity Approaches

The lexical-based approach is the simplest of these methods, with Cosine and Jaccard similarity being two commonly used techniques. Cosine similarity, a vector-based method defined in Eq. (1), struggles when sentences differ in length and fails to capture synonym relationships, such as distinguishing between "happy" and "joyful," unless enhanced by a vector representation that accounts for word meanings [6]. The Jaccard method, with its formula given in Eq. (2), on the other hand, counts only exact word matches, ignoring synonyms or words with similar meanings. It would be beneficial if this approach considered not only the exact words but also their meanings.

The remainder of the paper is structured as follows: Section 2 reviews the relevant literature. Section 3 outlines our proposed lexical similarity approach, focusing on the m-Jaccard algorithm and its features. Section 4 presents the results of our approach and a discussion of the findings. Finally, Section 5 concludes with a discussion of future directions for expanding our work.

II. REVIEW OF LITERATURE

Various methods for measuring sentence similarity have been studied extensively in the literature, aiming to enhance accuracy and quality and to address related challenges. In a recent study on lexical similarity, Ahmad and Faisal [2] introduced a hybrid string similarity method that combines

lexical features, word embeddings, and corpus statistics to evaluate sentence similarity. However, its effectiveness relies on access to specific datasets.

Yoo *et al.,* [8] proposed a deep learning model paired with a lexical-relationship-based string similarity approach, outperforming standalone deep learning models with a peak performance of 65%. Oussalah and Mohamed [9] examined semantic similarity not merely as a tool but by analyzing meaning from the word level to the sentence level.

Steck *et al.,* [4] stressed the need for careful application of cosine similarity in sentence similarity measurements, warning against its uncritical use. Several researchers have implemented sentence similarity techniques in practical applications, such as assessing subjective answers and grading student responses. Leacock and Chodorow [10] introduced C-rater, a system that uses paraphrase recognition for semantic content assessment.

Li *et al.,* [2] applied the K-nearest neighbor (KNN) classifier for automated essay scoring, utilizing a text categorization model based on the Vector Space Model. Nooralahzadeh *et al.,* [8] proposed a scoring method for free-text student responses [23] using a modified Bilingual Evaluation Understudy (M-BLEU) algorithm to identify the closest reference answer and generate a score. Kakkonen *et al.,* [12] developed an automatic essay grading system by comparing essays with learning materials using Latent Semantic Analysis (LSA), Probabilistic LSA (PLSA), and Latent Dirichlet Allocation (LDA).

Dhokrat *et al*., [14] suggested an evaluation system that uses a centralized file containing model answers and reference materials for each question. Islam and Hoque [10] proposed an automated essay grading system based on Generalized Latent Semantic Analysis (GLSA), incorporating word order and n-grams.

Ramachandran *et al.,* [15, 6] introduced a scoring technique for short answers using word-ordering graphs to detect patterns from rubrics and exemplary responses, while Sakaguchi *et al.,* [12] utilized word and character n-grams to extract features for content-based short-answer scoring. Surveys by Farouk [14] and Bounab *et al.,* [15] outlined various methods for string comparison, highlighting both sentence similarity approaches and the available datasets for these methods. Based on these insights, we propose a simple lexical sentence similarity method that can be integrated with hybrid approaches and tailored to various applications depending on specific needs and contexts.

III. OBJECTIVES

In our proposed method, we modify the Jaccard approach to emphasize word-level semantic meaning, naming it the modified Jaccard (m-Jaccard) algorithm. This approach not only compares word usage in a sentence but also considers the meanings of words by factoring in their synonyms. Keyword similarity differs from token or Jaccard similarity in that it focuses solely on the key terms within a sentence. In contrast, Jaccard-based n-gram methods consider all tokens or words semantically. This distinction often results in improved performance when using a lexical similarity approach.

IV. METHODOLOGY

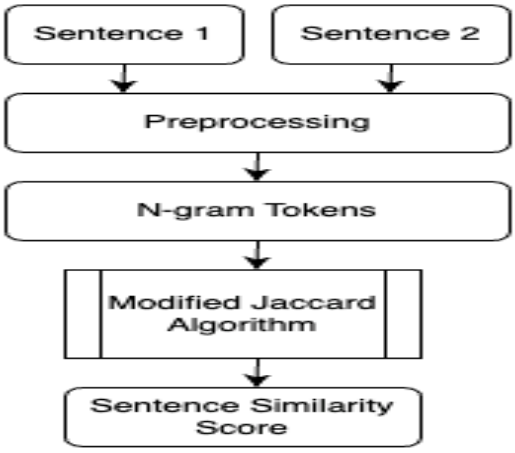

Fig.2 General Work flow of Proposed Method

The sentence similarity method primarily focuses on keywords. The main objective of this stage is to identify key terms from both Sentence 1 and Sentence 2 that will be compared to assess how similar the two sentences are. A Sentence 2 text containing the exact keywords, either lexically or semantically, as Sentence 1 will result in a higher similarity score. Algorithm: Modified Figure 2 illustrates that the sentence similarity function starts by splitting the two sentences into n-gram tokens, and all those tokens are compared lexically or semantically. Sentences with higher n-gram semantic similarity receive a higher similarity score [4]. For example, consider the following two sentences:

Sentence 1: “people love playing and watching cricket a lot for last twenty years.”
Sentence 2: “Individuals enjoy engaging and observing cricket extensively over the past two decades.”

Sample unigram, bigram, trigram, and n-gram matching examples:

1. Individuals – People
2. Individuals enjoy – People love
3. Individuals enjoy engaging – People love playing
4. Engaging and observing cricket – Playing and watching cricket
5. Individuals enjoy engaging and observing cricket – People love playing and watching cricket

Our proposed method begins with pre-processing steps, such as cleaning the text and splitting sentences into tokens to prepare them for further processing.

A. Jaccard Score

Input: Sentence1, Sentence2
Output: Returns Sentence Similarity Score
For keyword in R_k:
  For keyword in A_k:
    If $A_k[i] = R_k[i]$:
      A1 += 1 # if the keyword is same
    Else if $A_k[i] = R_k[\alpha(i)]$:
      A2 += 1 # if the keyword is same in case (i)
    Else if $A_k[i] = R_k[\beta(i)]$:
      A3 += 1 # if the keyword is same in case (ii)
    Else if $A_k[i] = R_k[\gamma(i)]$:
      A4 += 1 # if the keyword is same in case (iii)
union_Ak_Rk ← set(A_k) ∪ set(R_k) # Calculate the union of Ak and Rk Keywords
Return (A1 + A2 + A3 + A4) / len(union_Ak_Rk) # Return the similarity score

Algorithm 1 explains that the content similarly can be matched in three ways in which a word might be similar: synonymically (alpha), or base-wordly (beta), or quantitatively (gamma).

A meaningful word similarity result can be obtained by combining all three of these techniques, and when applied together, a semantically identical sentence will yield a perfect score. If a word does not fit into any of these categories-alpha, beta, or gamma-it can be considered irrelevant in the word similarity matching task.

For example, the sentences:
1. *Infants always enjoy funny tales.*
2. *Babies constantly love comic stories.*
3. *Old man rarely hates tragic events.*
4. *Enjoy funny tale.*

Jaccard method performs better than the standard Jaccard method, as shown in Figure 3.

TABLE I AN EXAMPLE SENTENCE – SIMILARITY SCORES

Methods	s1Vs2	s1Vs4	s2Vs4	s3Vs4
Jaccard	0.0	0.6	0.0	0.0
m-Jaccard	1.0	0.6	0.6	0.0
Hu-Eva	0.9	0.7	0.7	0.0

The example sentence similarity scores are presented in Table I. Sentences S1 and S2 are nearly identical, with a human evaluation score of 0.9. However, the Jaccard method assigns a score of 0.0. In contrast, our approach yields a perfect score of 1.0, indicating that the sentences are identical.

V. RESULTS AND DISCUSSION

A. Dataset Used

For the experiments, around 40 computer science students from the Central University of Tamil Nadu answered 100 questions on various computer science topics. Subject experts then evaluated these responses to establish ground truth, and this data is used for analysis in the results and discussion section of the paper.

B. Results of m-Jaccard

Anskey = "An algorithm is a step-by-step procedure or set of rules designed to solve a specific problem or perform a specific task." Table II presents sample scores for both the Jaccard and modified Jaccard methods. In this comparison, Sentence 1 is treated as the fixed answer key, while Sentence 2 represents the student's response. The focus of this approach is on content similarity, specifically word-level matching. Expert evaluations show that the modified

TABLE II A SAMPLE SET OF LEXICAL SIMILARITY SCORES OF JACCARD AND M-JACCARD

S. No.	Name	Responses	Ja-Sco	mJa-Sco	Hu-Sco
1	Stu 1	Step by step procedure to solve problem…	0.12000000	0.1545455	0.18
2	Stu 2	Algorithm is finite sequence of computation…	0.14285714	0.1471861	0.16
3	Stu 3	Algorithm is a step-by-step representation…	0.18181818	0.2107438	0.25
4	Stu 4	Algorithm is written for procedural approach…	0.19230769	0.3863636	0.56
5	Stu 5	A sequence of instruction of an input…	0.09090909	0.1404959	0.18
6	Stu 6	Algorithm is step by step process used…	0.11538462	0.1188811	0.15
7	Stu 7	Algorithm is used to do a process or work…	0.19230769	0.3566434	0.42
8	Stu 8	Algorithm is defining something…	0.10526316	0.1220096	0.12
9	Stu 9	It is sequences of steps to get an output…	0.18181818	0.2107438	0.24
10	Stu 10	It is nothing but the sequenced manner or…	0.11538462	0.1486014	0.19

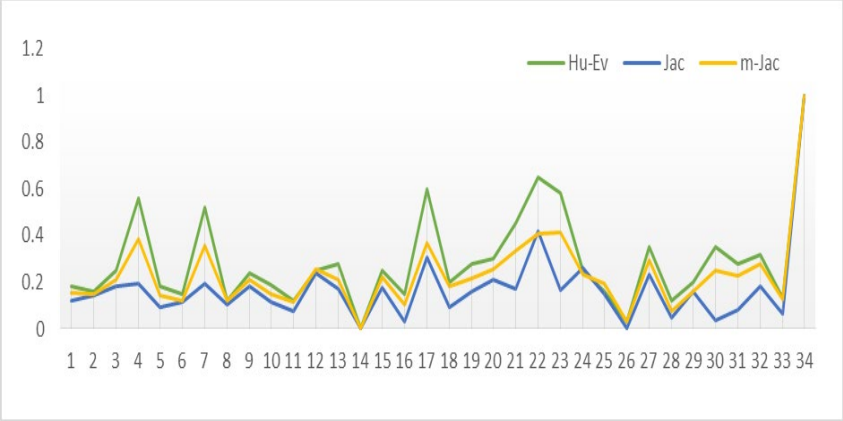

Fig.3 Performance Comparison of Jaccard and Modified-Jaccard

C. Performance of m-Jaccard

Table III summarizes the final sentence similarity scores of various lexical [22, 24] methods, and our proposed method surpasses some baseline approaches in sentence similarity results, as illustrated in Figure 4.

TABLE III LEXICAL SIMILARITY SCORE COMPARISON OF VARIOUS APPROACHES

Name	Human	Cosine	Jaccard	Proposed
Stu1	0.15	0.18	0.10	0.17
Stu2	0.25	0.15	0.09	0.18
Stu3	0.20	0.28	0.03	0.20
Stu4	0.15	0.14	0.06	0.15
Stu5	0.30	0.48	0.10	0.22
Stu6	0.25	0.36	0.19	0.26
Stu7	0.50	0.50	0.37	0.41
Stu8	0.15	0.12	0.04	0.15
Stu9	0.30	0.30	0.23	0.30
Stu10	0.15	0.12	0.05	0.15

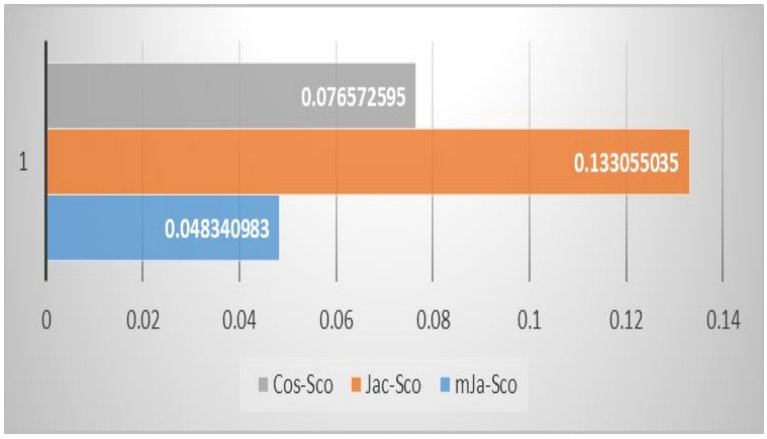

Fig.4 Score Variation of Lexical Similarity Methods

VI. CONCLUSION

Lexical-based N-gram similarity plays a crucial role in the concept of sentence similarity, and our proposed method can be used as a tool. Unlike advanced models [25], it delivers superior results and accuracy while requiring minimal resources. This approach can be seamlessly integrated into any hybrid sentence similarity model as needed. In our future work, we will concentrate on sentence similarity in the evaluation of students' subjective answers. To effectively assess subjective answers, additional approaches are required alongside lexical similarity, such as semantic similarity, keyword position order, contextual similarity, and others. Our next challenge will be concentrating on domain-specific and domain-adaptation [2A] concepts, such as subjective answer evaluation.

Declaration of Conflicting Interests
The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The authors received no financial support for the research, authorship, and/or publication of this article.

Use of Artificial Intelligence (AI)-Assisted Technology for Manuscript Preparation
The authors confirm that no AI-assisted technologies were used in the preparation or writing of the manuscript, and no images were altered using AI.

ORCID
P. Senthilkumar http://orcid.org/0009-0000-3881-5369
K. Nandhini http://orcid.org/0000-0003-4778-6525

REFERENCES

[1] Y. Li, D. McLean, Z. Bandar, J. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.

[2] F. Ahmad and M. Faisal, "A novel hybrid methodology for computing semantic similarity between sentences through various word senses," *International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 58–77, 2022.

[3] X. Yang *et al*., "Measurement of semantic textual similarity in clinical texts," *JMIR Medical Informatics*, vol. 8, no. 11, e19735, Nov. 2020.

[4] Z. H. Amur, M. T. H. Rahman, and M. H. A. Rahman, "Short-text semantic similarity (STSS): Techniques, challenges and future perspectives," *Applied Sciences*, vol. 13, no. 6, Art. 3911, Mar. 2023.

[5] M. Farouk, "Sentence semantic similarity based on word embedding and WordNet," in *Proceedings of the 13th International Conference on Computer Engineering and Systems (ICCES)*, Cairo, Egypt, 2018, pp. 33–37.

[6] H. Steck, C. Ekanadham, and N. Kallus, "Is cosine-similarity of embeddings really about similarity?" in *Companion Proceedings of the ACM Web Conference 2024*, 2024, pp. 887–890.

[7] T. Wang *et al*., "A joint FrameNet and element-focusing Sentence-BERT method of sentence similarity computation (FEFS3C)," *Expert Systems with Applications*, 2022.

[8] Y. Yoo, T.-S. Heo, Y. Park, and K. Kim, "A novel hybrid methodology of measuring sentence similarity," *Symmetry*, vol. 13, no. 8, p. 1442, 2021.

[9] M. Oussalah and M. Mohamed, "Knowledge-based sentence semantic similarity: algebraical properties," *Progress in Artificial Intelligence*, vol. 11, no. 1, pp. 43–63, 2022.

[10] C. Leacock and M. Chodorow, "C-rater: Automated scoring of short-answer questions," *Computers and the Humanities*, vol. 37, pp. 389–405, 2003.

[11] F. Nooralahzadeh *et al*., "Progressive transformer-based generation of radiology reports," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic, 2021, pp. 2824–2832.

[12] T. Kakkonen, N. Myller, E. Sutinen, and J. Timonen, "Comparison of dimension reduction methods for automated essay grading," *Journal of Educational Technology & Society*, vol. 11, no. 3, pp. 275–288, 2008.

[13] A. Dhokrat, H. Gite, and C. N. Mahender, "Assessment of answers: Online subjective examination," in *Proceedings of the Workshop on Question Answering for Complex Domains*, Mumbai, India, 2012, pp. 47–56.

[14] M. M. Islam and A. L. Hoque, "Automated essay scoring using generalized latent semantic analysis," in *Proceedings of the 2010 International Conference on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2010, pp. 358–363.

[15] L. Ramachandran, J. Cheng, and P. Foltz, "Identifying patterns for short answer scoring using graph-based lexico-semantic text matching," in *Proceedings of the 10th Workshop on Innovative Use of NLP for Building Educational Applications*, 2015, pp. 97–106.

[16] K. Sakaguchi, M. Heilman, and N. Madnani, "Effective feature integration for automated short answer scoring," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*, 2015, pp. 1049–1054.

[17] M. Farouk, "Measuring sentences similarity: A survey," *Indian Journal of Science and Technology*, vol. 12, no. 25, pp. 1–11, 2019.

[18] Y. Bounab *et al*., "Sentence to sentence similarity: A review," in *Proceedings of FRUCT'25 (Finnish–Russian University Cooperation in Telecommunications)*, Helsinki, Finland, Nov. 2019.

[19] B. Li, J. Lu, J.-M. Yao, and Q.-M. Zhu, "Automated essay scoring using the KNN algorithm," in *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, 2008.

[20] R. A. A. Akinyemi, W. Ajayi, and A. Atuman, "Automation of customer support system (Chatbot) to solve web-based financial and payment application service," *Asian Journal of Computer Science and Technology*, vol. 12, no. 2, pp. 1–17, 2023.

[21] A. G. L. Raja, F. S. Francis, and P. Sugumar, "Construction of lexicons to perk up re-clustering," *Asian Journal of Computer Science and Technology*, vol. 7, no. 3, pp. 82–85, 2018.

[22] S. J. Lakshmi and M. Thangaraj, "Recommender system for student performance using EDM," *Asian Journal of Computer Science and Technology*, vol. 7, no. 3, pp. 53–57, 2018.

[23] R. K. Jain, "A survey on different approach used for sign language recognition using machine learning," *Asian Journal of Computer Science and Technology*, vol. 12, no. 1, pp. 11–15, 2023.

[24] V. Bonta, N. Kumaresh, and N. Janardhan, "A comprehensive study on lexicon-based approaches for sentiment analysis," *Asian Journal of Computer Science and Technology*, vol. 8, no. S2, pp. 1–6, 2019.

[25] A. Ahmadi, "Unravelling the mysteries of hallucination in large language models: Strategies for precision in artificial intelligence language generation," *Asian Journal of Computer Science and Technology*, vol. 13, no. 1, pp. 1–10, 2024.