Alkadhim Journal for Computer Science, Vol. 3, No. 1 (2025)

Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

IRAQI Academic Scientific Journals

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

KJCS ALKADHIM JOURNAL FOR COMPUTER SCIENCE

A Comprehensive Framework for Quality Assurance of Generative AI Text

[1]Yasmin Makki Mohialden*, [1]Nadia Mahmood Hussien, [2]Saba Abdulbaqi Salman

[1] Department of Computer Science, Collage of Science, Mustansiriyah University, Baghdad – Iraq

[2]Department of Computer Science, College of Education, Al-Iraqia University, Baghdad – Iraq

Article information

Article history:
Received: Nov, 02, 2024
Accepted: Mar, 16, 2025
Available online: Mar, 25, 2025

Keywords:
Generative AI,
Quality Assurance (QA),
Text Generation,
ChatGPT,
Natural Language Processing

Corresponding Author:
Yasmin Makki Mohialden
ymmiraq2009@uomustansiriyah.edu.iq

DOI:
https://doi.org/10.61710/kjcs.v3i1.84

Abstract

This research proposes a rigorous Quality assurance (QA) framework for AI-generated text to evaluate and ensure content quality. Multiple dimensions are assessed, including grammar, spelling, prompt relevance, and linguistic diversity. Python-based language models are used to find grammatical errors, TF-IDF vectorization with cosine similarity to judge relevance, and natural language processing (NLP) to look at lexical variety. This methodology incorporates these criteria into a unified and consistent evaluation process to assure high-quality text output, unlike many other QA methods. The framework tackles major industry issues. It enhances the accuracy and readability of medical reports, guaranteeing the effective communication of crucial information. It helps lawyers write clear, unambiguous documents. It helps educators create contextually relevant, engaging learning materials that promote understanding and interaction. Flexibility and scalability allow the framework to meet different user demands across disciplines. An actual implementation shows that the framework improves AI-generated content over previous methods. Higher relevance, grammatical identification, and lexical diversity are key benefits. Actionable feedback allows generative AI systems to be refined and model performance improved. Possible inclusions include real-time QA systems, domain-specific relevance models for specialized domains, and multimodal QA that evaluates text, pictures, and audio. This method is reliable, morally sound, and useful because it takes into account all the important quality factors. It makes it possible for new and advanced generative AI applications in many areas.

1 Introduction

Computational methods that can produce seemingly original, relevant content—like writing, images, or audio—from information used for training are called generative AI. The broad popularity of this innovation, exemplified by products like GPT-4, Copilot, and Dall-E 2, is presently transforming how we collaborate and conduct business. In addition to being utilized creatively to produce fresh text that mimics authors or new visuals that mimic artists, creative AI systems can and will help humans by acting as intelligent question-answering systems. Solutions in this context comprise IT help desks, where generative AI assists with routine chores like food preparation, healthcare recommendations, and transitional expertise duties [1]. Although little has been learned about how new AI technologies may affect workers' performance and learning, this could significantly influence [2]. According to industry projections, generative AI has the potential to increase global GDP by 7% while also replacing 300

million specialists. Undoubtedly, this has broad repercussions not only regarding the company & Systems Engineering (BISE) society, where we will encounter revolutionary opportunities but also challenges and risks that must be addressed to guide the ethical and sustainable use of electronic systems [1]. Previous studies lack comprehensive evaluative frameworks that simultaneously address grammar, relevance, and diversity.

The primary problem addressed in this paper is the lack of a comprehensive and automated framework for evaluating the quality of text generated by AI models. Existing approaches often focus on isolated aspects, such as grammatical correctness or contextual relevance, but rarely integrate multiple quality dimensions into a unified evaluative framework. This fragmented approach might overlook quality issues, resulting in grammatically accurate but contextually irrelevant or lexically repetitive content. We bridge this gap by integrating multiple QA metrics into a unified system. This approach evaluates content grammar, spelling, input prompt relevance, and linguistic variation. Our comprehensive QA framework addresses multiple quality variables simultaneously, providing a more robust evaluation mechanism. The advanced tools used in our system include Python for grammatical checks, TF-IDF vectorization with cosine similarity for relevance assessment , and NLTK for lexical diversity analysis [3, 4]. This framework has several uses. It ensures accurate and comprehensible medical reporting in healthcare. Legal technology can help draft transparent papers. The framework helps create informed and exciting educational content. Our methodology enhances AI-generated outputs in these critical domains by providing a holistic QA mechanism.

The paper describes an innovative and comprehensive method for ensuring the quality of generative AI text output. Our proposed method solves high-stakes application difficulties by concurrently addressing major quality parameters and enhancing generative AI quality evaluation.

Text generative AI's generic implications and applications are:

- Model Improvement: QA metrics provide actionable feedback for refining generative AI models, guiding iterative development efforts to enhance text quality over time.
- Content Filtering: QA scores can assist in filtering or post-filtering generated content, ensuring that only high-quality outputs are presented to users or downstream application
- Domain-specific Applications: Tailoring QA metrics to specific domains or applications enables customized evaluation criteria aligned with user requirements and expectations.

This study proposes an integrated approach to quality assurance for generative AI models, demonstrating improved results in grammar detection, relevance scoring, and lexical diversity [5-7].

2 Literature Review

AI model text quality assurance (QA) has recently been extensively studied. Several ways to assess and improve text quality include grammatical accuracy, relevancy, and linguistic variety. In [3], they introduced the GPT-3 model, which significantly advanced the field of natural language generation by demonstrating few-shot learning capabilities. However, the study primarily focused on the model's performance in generating coherent and contextually relevant text without a comprehensive framework for quality assurance.

In this paper [8], they develop a novel approach called categorical line generative adversarial network (CS-GAN), combining adversarial generative networks, recurrent neural networks, and reinforcement learning methodologies. The suggested model can provide category sentences that expand the initial data set and improve the supervised training process's applicability. They assess how well CS-GAN performs in sentiment evaluation. The accuracy enhancement in polarity recognition is demonstrated through qualitative testing on a tiny data set with high classification detail. In [9], Researchers report on fieldwork now being conducted to ensure the quality of patent-generated dialogue systems taught for online interviews. The action team identified 38 criteria, of which Fifteen were particularly relevant to the developing solution and for which they created automated test cases. Their findings show six test case designs can identify significant variations across potential models. Although the task of natural language processing programs' quality assurance is complicated, they offer the first steps into an

automated framework for artificial intelligence decision-making inside the framework of a talking agent that is always changing. In [4], OpenAI released a technical report on GPT-4, highlighting improvements in text generation quality and the introduction of mechanisms to reduce biases and improve factual accuracy. Despite these advancements, the report did not address a unified approach for evaluating multiple quality dimensions simultaneously [5]. In [10], Weakeners are classified into four types of unpredictability according to classification: aleatory, epistemic, ontology, and argumentative. Additionally, it divides techniques for management into three categories: prevention, identification, and presentation. Finding reasoning ambiguity or defeaters is a crucial step in building more robust guarantee cases. We investigate the potential of OpenAI's GPT-4 Turbo, a powerful large languages model, for streamlining this entire procedure. We concentrate on how it might be used to identify defeaters in assuring instances that are expressed in Eliminative Argumentation (EA) notation. Our preliminary analysis evaluates GPT-4 Turbo's ability to comprehend and utilize this notation, which is essential for producing defeaters. The outcomes show how proficient GPT-4 Turbo is in EA writing and how it can produce a wide variety of defeaters to improve the strength and dependability of assurance cases. Tabel 1 illustrates the comparison of the Proposed Method with Previous Work:

Table 1: Comparison of proposed method with previous work

Aspect	Related Work [3]	Related Work [8]	Related Work [9]	Related Work [4]	Related Work [10]	Proposed Method
Focus	Few-shot learning capabilities of GPT-3	Categorical line GAN for sentiment analysis	Quality assurance in patent dialogue systems	Improvements in text generation quality in GPT-4	Defeaters in assurance cases using GPT-4 Turbo	Comprehensive QA framework for AI-generated text
Methodology	Performance in generating coherent, contextually relevant text	Combination of GANs, RNNs, and RL	Automated test cases for quality assurance	Mechanisms to reduce biases and improve factual accuracy	Identifying defeaters in EA notation using GPT-4 Turbo	Multiple metrics, including grammar, relevance, and linguistic diversity
Evaluation Metrics	Coherence and contextual relevance	Polarity recognition accuracy	Variations across models using test cases	Text generation quality, bias reduction	Proficiency in EA notation and variety of defeaters	Grammar, spelling, relevance, lexical diversity
Strengths	Significant advancement in natural language generation	Improved applicability in supervised training	Automated framework for AI decision-making	Introduction of mechanisms to improve factual accuracy	Streamlining defeater identification process	Robust mechanism ensuring generated text meets quality standards
Weaknesses	Lack of comprehensive QA framework	Evaluation on a small data set	Complexity in NLP quality assurance	No unified approach for evaluating multiple quality dimensions	Preliminary analysis, specific to EA notation	Easily extendable and customizable for various applications
Tools/Technologies	GPT-3	GAN, RNN, RL	Automated test case design	GPT-4	GPT-4 Turbo, EA notation	python, TF-IDF vectorization, NLTK

3 Methodology

Figure 1 depicts the suggested solution structure for assurance of quality (QA) of AI-generated text. The process commences with an entry of created text that is subsequently examined for grammatical and syntax issues utilizing the Language Tool package. Meanwhile, the text's significance is appraised using TF-IDF vectorization and cosine correlation to get an importance score. Furthermore, the algorithm uses the NLTK library to calculate lexical variety and generate a score. The outcomes of these tests are then merged into an integrated QA report, which is printed as the concluded QA report, guaranteeing that the text meets the quality requirements.

The QA framework integrates Python libraries for error detection, TF-IDF for relevance scoring, and NLTK for lexical diversity evaluation. Figure 1 show how Break down each component (e.g., grammatical analysis, scoring mechanisms) and clarify its purpose.

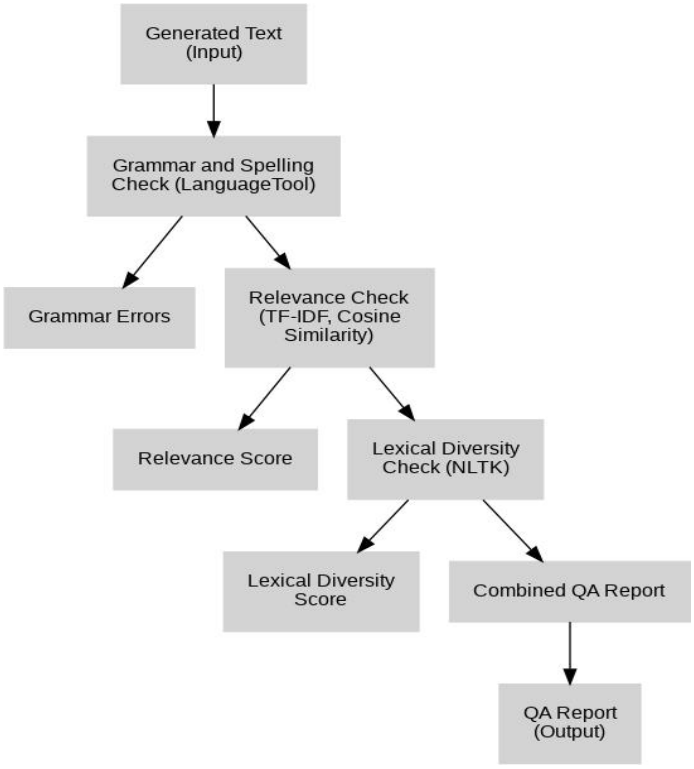

Figure 1: The block diagram of the proposed Method

Tables 2 and 3 illustrate the functional requirements and non-functional requirements

Table 2: Functional Requirements of the proposed Method

Requirement	Description
Grammar and Spelling Check	- Ensure detection of grammatical errors. - Identify spelling mistakes. - Provide a list of detected errors and their types.
Relevance Check	- Compute the relevance score between generated text and reference text. - Utilize TF-IDF and cosine similarity for relevance assessment.
Lexical Diversity Check	- Calculate the lexical diversity score of the generated text. - Consider the ratio of unique words to total words.
Combined QA Report	- Compile all QA metrics into a comprehensive report. - Present the report in a readable and understandable format.

Table 3: Non-Functional Requirements of the proposed method

Requirement	Description
Performance	- Ensure efficient processing of text inputs. - Optimize algorithms for scalability and speed.
Accuracy	- Maintain high accuracy in detecting grammar and spelling errors. - Ensure precise computation of relevance and lexical diversity scores.
Robustness	- Handle various text inputs, including various lengths and complexities. - Implement error-handling mechanisms to manage unexpected inputs.
Usability	- Design a user-friendly interface for easy interaction with the QA framework. - Provide clear instructions for usage and interpretation of QA reports.

The suggested QA infrastructure is built on such variables and description of parameters and features, as in Tables 4 and 5 sequentially, which ensure that the AI-generated language is correct in grammar, related to the specified prompted or comparison text, and verbally unique.

Table 4: Description of Parameters

Parameter	Description
Generated text	The text generated by the AI model.
Reference text	The reference text for comparison.

Table 5: General description of features

Feature	Description
Grammar and Spelling Check	Utilizes Language Tools for detecting grammatical and spelling errors.
Relevance Check	Computes TF-IDF vectors and cosine similarity for relevance assessment.
Lexical Diversity Check	Tokenizes text, calculates unique words, and computes diversity ratio.

The output of the proposed method is a comprehensive quality assurance report encompassing the metrics presented in Table 6. Table 6 demonstrates the correlation between the relevance scores and lexical diversity, elucidating model consistency.

Table 6: QA Metrics report

Metric	Description
Grammar Errors	Several grammatical errors were detected in the generated text.
Error Types	Types of grammatical errors identified (e.g., spelling mistakes, punctuation errors).
Relevance Score	The cosine similarity score indicates the relevance of the generated text to the reference text.
Lexical Diversity Score	The ratio of unique words to total words in the generated text.

4 Discussion of Metrics

4.1 Grammar Error Rate

This metric identifies grammatical errors present in the generated text, A lower number of grammar errors reflects higher linguistic accuracy, improving both the readability and credibility of the text[14].

$$GER = (\text{Number of Grammar Errors} / \text{Total Words}) \times 100 \quad (1)$$

4.2 Error Types

This refers to the specific grammatical issues detected in the text, such as spelling, punctuation, and syntax errors, categorizing these errors helps users prioritize corrections and refine their model feedback for future text generations.

5 Relevance Score(RS)

The relevance score quantifies the contextual alignment between the generated text and the reference text or prompt, A higher relevance score indicates better adherence to the intended topic or context, demonstrating the model's ability to produce coherent and on-topic responses.

$$RS = \cos(\theta) = (A \cdot B) / (\|A\| \|B\|) \quad (2)$$

- **Term Frequency (TF):** The ratio of the number of times a term appears in a document to the total number of terms in that document.

- **Inverse Document Frequency (IDF):** The logarithm of the total number of documents divided by the number of documents containing the term, assessing the rarity of the term across the corpus.

where: A = TF-IDF vector of generated text B = TF-IDF vector of reference text[15].

5.1 Lexical Diversity Score((LDS)

This metric measures the variety and richness of vocabulary in the text, A higher lexical diversity score reflects a broader range of words and expressions, enhancing the text's expressiveness and reducing repetitiveness[16].

$$LDS = (\text{Number of Unique Words} / \text{Total Words}) \quad (3)$$

5.1 Challenges

6.1.1 Scalability Issues

Processing large datasets can strain resources, affecting performance and scalability.

5.1.2 Dataset Diversity

Limited dataset diversity may impair the model's ability to assess relevance comprehensively, particularly in nuanced contexts.

6 Results and Discussions

Table 7 presents the findings of a Quality Assurance (QA) evaluation for AI-generated text across various applications, utilizing the proposed QA system. The Proposed Method assesses key aspects such as grammar, relevance, and lexical diversity, providing a comprehensive report of the text quality.

Table 7: the findings of a Quality Assurance (QA) examination for AI-generated text

AI Application	Text	Results	date	Reference
Gatgpt	Software engineering is a systematic, disciplined, and quantifiable approach to the development, operation, and maintenance of software. It involves the application of engineering principles to software creation to ensure that the software is reliable, efficient, and meets user requirements.	Grammar Errors: 0 Error Types: Relevance Score: 0.49 Lexical Diversity Score: 0.72 Quality Assurance Report: Grammar Errors: 0 Error Types: Relevance Score: 0.49318595541091415 Lexical Diversity Score: 0.717948717948718	31/5/2024	What is software engineering? Software engineering is an engineering discipline that is concerned with all aspects of software production from initial conception to operation and maintenance [11]

Cybersecurity	Cybersecurity is a state, political, and even military problem. However, this is an area where – thanks to individual skills and devoted time – to a certain extent, specialist knowledge, information, and experience can be acquired by people who do not belong to the "formal" expert circles. Many techniques and methods in cybersecurity were developed and used by various "interest groups". W	Grammar Errors: 0 Relevance Score: 0.06 Lexical Diversity Score: 0.94 Quality Assurance Report: Grammar Errors: 0 Error Types: Relevance Score: 0.059567017973396294 Lexical Diversity Score: 0.9411764705882353	31/5/2024	Cybersecurity is a state, political, and even military problem. However, this is an area where – thanks to individual skills and devoted time – to a certain extent, specialist knowledge, information, and experience can be acquired by people who do not belong to the "formal" expert circles. Many techniques and methods in cybersecurity were developed and used by various "interest groups" [12].

6.1 Interpretation of Results

6.1.1ChatGPT

- Text: The generated content describes software engineering.
- Results:
  - Grammar Errors: 0 (indicating no grammatical mistakes).
  - Relevance Score: 0.49 (moderate relevance to the reference text).
  - Lexical Diversity Score: 0.72 (fairly diverse vocabulary use).
- Quality Assurance Report:
  - Grammar Errors: Confirmed as 0.
  - Relevance Score: 0.4932 (slightly more precise value).
  - Lexical Diversity Score: 0.7179 (slightly more precise value).
- Date: QA performed on 31/05/2024.
- Reference: The reference text is about software engineering, citing Sommerville, I. (2016), aligning with the generated content [13].

6.1.2 Cybersecurity

- Text: The generated content discusses cybersecurity challenges and the state of the field.
- Results:
  - Grammar Errors: 0 (indicating no grammatical mistakes).
  - Relevance Score: 0.06 (low relevance to the reference text).
  - Lexical Diversity Score: 0.94 (high vocabulary diversity).

- Quality Assurance Report:
    - Grammar Errors: Confirmed as 0.
    - Relevance Score: 0.0596 (slightly more precise value).
    - Lexical Diversity Score: 0.9412 (slightly more precise value).
- Date: QA performed on 31/05/2024.
- Reference: The reference text is about cybersecurity [12].

6.1.3 Analysis

6.1.3.1 Grammar Errors

Both texts demonstrate flawless grammar, suggesting high linguistic accuracy.

6.3.1.2Relevance Score

The software engineering text exhibits moderate alignment with the reference text (0.49), while the cybersecurity text shows low relevance (0.06).

6.1.1.3 Lexical Diversity Score

The cybersecurity text (0.94) employs a more varied vocabulary compared to the software engineering text (0.72), suggesting greater linguistic richness.

6.2 Evaluation of Proposed Method Efficacy and Reliability

Text quality is evaluated systematically when the proposed QA approach is applied to generative AI text outputs. Actionable metrics include:

6.2.1 Grammar Issues

- Grammar errors, while absent here, can indicate limitations in the AI model's language generation capabilities.
- Improvements may involve post-processing techniques or model fine-tuning.

6.2.2 Relevance Score

- A low relevance score reflects a lack of topical alignment.
- Enhancements: Refine training datasets, adjust model parameters, and optimize prompt design to improve context adherence.

6.2.3 Lexical Diversity Score

- A lower score may result in monotonous text.
- Recommendations: To improve vocabulary, add diverse content to training data and introduce diversity-promoting objectives during model training.

7 Conclusion

The proposed generative AI text output QA assesses text quality in several ways. The framework includes grammar, spelling, relevance, and linguistic diversity tests to help users assess generated text quality for diverse applications. This paradigm evaluates grammar, contextual relevance, and linguistic richness. Consumers may make educated decisions and develop generative AI models using the QA report's comprehensive input.The recommended QA technique improves generative AI text outputs and adds value across applications and domains. Possible Career: 1. Advanced Relevance Assessment: Researchers may explore semantic similarity metrics or domain-specific relevance models. Some domains and contexts may benefit from these relevance score accuracy and robustness methodologies. 2. Analyzing dynamic quality levels based on use situations or user preferences may make the QA framework more adaptable. The framework fulfills application and consumer demands by allowing users change quality criteria. 3. Multimodal QA: Visuals or sounds may improve multimodal generative

AI system evaluations. Text, image, and audio analysis may increase output quality and coherence. 4. QA metrics-based real-time feedback systems may enable adaptive model refinement and continuous quality monitoring. Such techniques may enable generative AI systems adapt and modify during text generation by providing immediate feedback. 5. Human-in-the-loop QA: Expert annotations or crowdsourced evaluations can enhance automatic QA. Human comments can validate complex text quality aspects that are hard to algorithmically collect. Advanced QA improves generative AI text output reliability, usefulness, and effect across applications and domains. Semantic analysis models and real-time feedback are adaptive quality improvement priorities.

Acknowledgement: The authors would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq) and Al-Iraqia University in Baghdad, Iraq, for its support in the present work.

Conflict of Interest: The authors declare that there are no conflicts of interest associated with this research project. We have no financial or personal relationships that could potentially bias our work or influence the interpretation of the results.

References

[1] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, "Generative AI," *Business & Information Systems Engineering*, vol. 66, no. 1, pp. 111–126, 2024.

[2] E. Brynjolfsson, D. Li, and L. Raymond, "Generative AI at work," Cambridge, MA, 2023.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[4] OpenAI, "GPT-4," Mar. 14, 2023. [Online]. Available: https://openai.com/index/gpt-4-research/. [Accessed: Dec. 25, 2024].

[5] M. T. Baldassarre, D. Caivano, B. Fernandez Nieto, D. Gigante, and A. Ragone, "The social impact of generative AI: An analysis on ChatGPT," in *Proc. 2023 ACM Conf. Inf. Technol. for Social Good*, Lisbon, Portugal, Sep. 2023, pp. 363–373.

[6] T. K. Chiu, "The impact of Generative AI (GenAI) on practices, policies and research direction in education: a case of ChatGPT and Midjourney," *Interactive Learning Environments*, pp. 1–17, 2023.

[7] N. R. Mannuru, S. Shahriar, Z. A. Teel, T. Wang, B. D. Lund, S. Tijani, C. O. Pohboon, D. Agbaji, J. Alhassan, J. Galley, and R. Kousari, "Artificial intelligence in developing countries: The impact of generative artificial intelligence (AI) technologies for development," *Information Development*, p. 02666669231200628, 2023.

[8] Y. Li, Q. Pan, S. Wang, T. Yang, and E. Cambria, "A generative model for category text generation," *Information Sciences*, vol. 450, pp. 301–315, 2018.

[9] M. Borg, J. Bengtsson, H. Österling, A. Hagelborn, I. Gagner, and P. Tomaszewski, "Quality assurance of generative dialog models in an evolving conversational agent used for Swedish language practice," in *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, Pittsburgh, PA, May 2022, pp. 22–32.

[10] K. Khakzad Shahandashti, "Examining the effectiveness of generative artificial intelligence for the identification of defeaters in assurance cases," M.S. thesis, York University, Toronto, Ontario, 2024.

[11] I. Sommerville, “Software Engineering,” 10th ed. Boston, MA: Pearson Education Limited, 2016.

[12] L. Olejnik and A. Kurasiński, “Philosophy of Cybersecurity,” CRC Press, 2023.

[13] M. T. Younis, N. M. Hussien, Y. M. Mohialden, K. Raisian, P. Singh, and K. Joshi, "Enhancement of ChatGPT using API Wrappers Techniques," *Al-Mustansiriyah Journal of Science*, vol. 34, no. 2, pp. 82–86, 2023.

[14] Bryant, Christopher, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. "Grammatical error correction: A survey of the state of the art." *Computational Linguistics* 49, no. 3,pp. 643-701 , 2023.

[15] Widianto, Adi, Eka Pebriyanto, Fitriyanti Fitriyanti, and Marna Marna. "Document Similarity using Term Frequency-Inverse Document Frequency Representation and Cosine Similarity." *Journal of Dinda: Data Science, Information Technology, and Data Analytics* 4, no. 2 pp.149-153,2024.

[16] Kyle, Kristopher, Hakyung Sung, Masaki Eguchi, and Fred Zenker. "Evaluating evidence for the reliability and validity of lexical diversity indices in L2 oral task responses." *Studies in Second Language Acquisition* 46, no. 1 , pp. 278-299,2024.