

# Automated Scientific Article Generator System Based on GPT Algorithm

I Wayan Budi Sentana<sup>1</sup>, Ni Putu Eka Apriyanthi<sup>2</sup>, Ni Nyoman Harini Puspita<sup>3</sup>

## Abstract

Composing scientific articles and synthesizing relevant literature are often time-consuming and challenging tasks for researchers. This study developed an automated scientific article generator system that leverages advanced Artificial Intelligence capabilities to address these inefficiencies. The proposed system integrates the OpenAI API using the GPT algorithm to construct natural language generation with Web Scraping techniques, specifically targeting academic databases such as IEEE Xplore, to dynamically retrieve and incorporate up-to-date scholarly references. The system is designed to streamline the article writing process by generating cohesive, structured text based on user-defined topics and seamlessly embedding pertinent citations. The performance of the generated articles was rigorously evaluated using quantitative metrics: ROUGE (for lexical overlap) and BERTScore (for semantic similarity) against reference texts. Empirical results are highly promising: the system achieved a BERTScore F1-Score of 84.46%, demonstrating superior semantic correspondence and contextual relevance while extracting critical information from source texts. This proposed technique can be a potential solution to enhance writing efficiency and support academic documentation.

## Keywords:

Scientific Article Generation, GPT, Web Scraping, ROUGE, BERTScore

*This is an open-access article under the [CC BY-SA](#) license*



## 1. Introduction

Scientific writing stands as a fundamental pillar for the advancement of science and technology, serving as the primary vehicle for documenting research findings, expanding academic discourse, and advancing theoretical development [1]. Through the dissemination of scientific articles, researchers contribute critical insights to a global audience, establishing a valuable foundation for data-driven decision-making and serving as essential references for subsequent investigations. Consequently, the practice of scholarly writing plays a crucial role in technological development, scientific progress, and sustainable development.

In the academic sphere, scientific articles represent the standard format for presenting research concisely and systematically within official journals. These documents typically adhere to the IMRaD (Introduction, Methods, Results, and Discussion) structure, focusing on a specific research problem, which differentiates them from more extensive works such as theses or dissertations [2]. However, the process of authoring high-quality scientific articles is often fraught with challenges. The escalating demand for publishable material often encounters significant productivity hurdles, most notably the difficulty in identifying and synthesizing relevant literature. This activity alone is notoriously time-consuming, particularly for novice researchers [3]. Furthermore, factors such as low reading comprehension, grammatical ambiguity, and inadequate motivation can further impair a writer's efficiency [4].

**Corresponding Author:** Ni Putu Eka Apriyanthi ([epriyanthi@pnb.ac.id](mailto:epriyanthi@pnb.ac.id))

<sup>1</sup> I Wayan Budi Sentana, Politeknik Negeri Bali, Indonesia

<sup>2</sup> Ni Putu Eka Apriyanthi, Politeknik Negeri Bali, Indonesia

<sup>3</sup> Ni Nyoman Harini Puspita, Politeknik Negeri Bali, Indonesia

The rapid evolution of information technology offers robust solutions to mitigate these academic challenges. Data acquisition, essential for comprehensive information retrieval, can be efficiently managed using automated web scraping techniques [5]. Web scraping is an instrumental method for programmatically collecting vast amounts of data from online sources for immediate analysis or reference. Utilizing this technique allows writers to bypass manual searches, gaining expedited access to scholarly articles and other critical information, thereby contributing directly to the quality and timeliness of the final scientific article.

Furthermore, the rapid progress of Artificial Intelligence (AI) technology has created numerous opportunities to directly address obstacles in scholarly writing [6]. The Generative Pretrained Transformer (GPT), a transformer-based machine learning model accessed via the OpenAI API, represents a significant breakthrough[7]. This model is highly proficient at generating high-quality text, discerning context, imposing structural integrity, and performing automated grammatical corrections [8]. To support the scientific writing process, web scraping is utilized to obtain information and references from various online sources. The collected information from web scraping serves as a supporting reference for GPT, which then uses the OpenAI API to compose a complete scientific article with its references [9]. GPT has demonstrated immense potential in assisting with scientific writing, capable of generating accurate text with superior structure and grammar [10], and is highly effective in simplifying the creation of initial drafts, a process traditionally known to consume substantial time [11].

Based on this background, this study aims to develop an automated scientific article generator system with distinct features and comprehensive performance justification. The novelty of this research lies in creating a unified, end-to-end framework capable of real-time reference ingestion via Web Scraping from IEEE Xplore and subsequent context-aware text synthesis into a fully structured academic draft. The OpenAI GPT 3.5 Turbo API was chosen due to its superior capability in handling complex, constraint-based prompt engineering and its proven efficiency in generating semantically cohesive text from multiple, unstructured sources[12]. This choice allows the system to achieve high semantic understanding, which is critical for the academic synthesis task. The ultimate objective is to provide a robust, automated solution that demonstrably reduces the time consumption associated with manual literature review and article drafting, contributing significantly to academic efficiency.

## 2. Related Works

The development of automated text generation has emerged as a significant area of research, particularly following the advent of Large Language Models (LLMs). Various studies have explored the application of these models to address persistent challenges in scientific writing, document processing, and other specialized academic tasks. This section reviews key literature that provides the foundational and comparative context for our current research.

The theoretical backbone of modern LLMs, including the GPT architecture, stems from the "Attention Is All You Need" paper by Vaswani et al. [13]. This seminal work introduced the Transformer architecture and its core self-attention mechanism. This innovation fundamentally revolutionized Natural Language Processing by enabling models to assign varying weights to words within a sequence, thus fostering deeper contextual understanding and the generation of highly coherent text. Our proposed system is built upon this fundamental framework, leveraging the power of attention to generate contextually relevant content from disparate sources[14].

Research in the Indonesian context has widely explored the practical applications of generative models. D. H. Putra and L. R. Bahasuan [15] demonstrated the implementation of GPT 2 for generating abstracts from journal documents, validating the models' potential to automate specific components of the academic writing process. Similarly, Yuniati et al. [8] analyzed the performance of GPT 3 in content extraction using rigorous quantitative metrics such as ROUGE and BERTScore [16]. This methodology is highly relevant as it provides a standardized approach for evaluating the quality of text generated by a GPT model, which we directly adopt in our system's performance assessment.

Furthermore, the utilization of AI has been directed toward solving domain-specific problems. For instance, Rahayu [10] provided a general discussion on the utilization of AI in scientific writing, emphasizing its potential to streamline the drafting process and improve efficiency. Complementing this, Satriajati et al. [5], explored the critical role of Web Scraping in gathering and synthesizing large volumes of online information, a technique that is integral to our system for the dynamic retrieval of up-to-date scholarly references.

Recent studies further illustrate the growing trend of integrating LLM APIs into diverse technological solutions. Saputra et al. [17] successfully implemented a web-based ChatGPT summarizer, which shares a core function of synthesizing information with our system. The versatility of the API has also been shown through its integration into alternative platforms, such as a WhatsApp bot developed by Alia et al. [6] and its application in domain-specific tasks, including generating e-commerce product descriptions [9] and supporting restaurant applications [18]. These works collectively validate the necessity of combining generative power with dedicated system development.

More directly related to scientific document processing, Mu'alifah and Haviana developed a system for scientific article summarization utilizing GROBID for information extraction and a SciBERT-based LLM for generating summaries from uploaded PDF documents [19]. Their research successfully demonstrated the efficacy of combining specialized tools for data extraction with contextual models for text condensation. However, the study focuses on the summarization of already provided single documents, which addresses the efficiency of literature reading but not the entire synthesis process.

Our research extends these prior works by uniquely integrating the generative power of the OpenAI API with dynamic data retrieval capabilities via Web Scraping into a single, cohesive, and structured system. While preceding studies have often compartmentalized these aspects, focusing on mere summarization [19] or API implementation [18] our work combines them to create a comprehensive, end-to-end solution for generating structured scientific article drafts with embedded, relevant citations.

### 3. Proposed Method

The proposed system is designed with a modular architecture to ensure scalability and flexibility. This architecture comprises several key components that collaboratively facilitate the automated generation of scientific articles. The User Interface (UI) module provides an intuitive environment for users to input research topics, relevant keywords, specify desired article formats, and ultimately view the generated output. Concurrently, the Reference Collection Module (Web Scraping Module) is responsible for the automated retrieval of scientific article metadata, including titles, authors, abstracts, publication years, DOIs, and publishers, from online academic databases, specifically IEEE Xplore, based on user-provided keyword queries. These collected references are then systematically stored within a Database Management Module to house structured information from the scraped articles, ensuring efficient access and seamless integration into the subsequent text generation process.

At the core of the system lies the Text Generation Module (OpenAI API Module), which interfaces with the OpenAI API (specifically the GPT-3.5 Turbo model) to generate scientific article drafts. This crucial process involves meticulous, prompt construction that integrates

both user-defined topics and information derived from the retrieved references. Finally, a Text Preprocessing Module ensures the cleaning and formatting of both the input text and the output from the OpenAI API, aligning them with the desired structure and academic writing style of scientific articles. A comprehensive system architecture diagram is presented in Fig. 1.

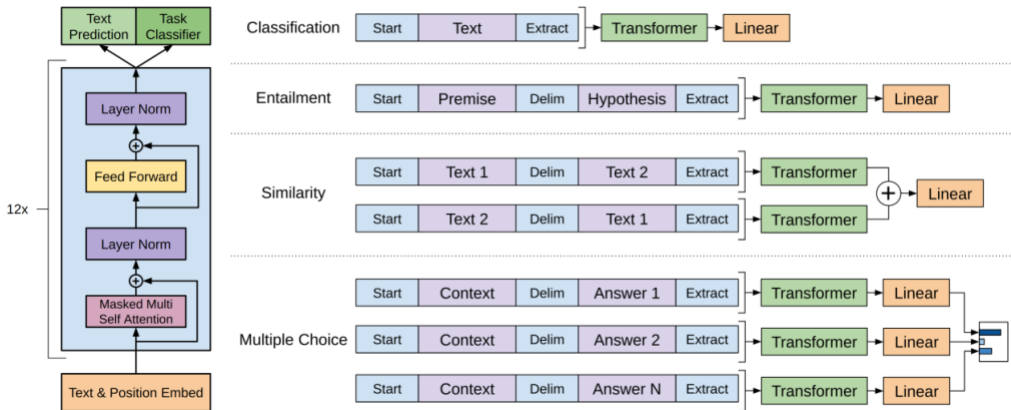


Fig. 1 GPT Architecture

### 3.1 System Architecture and Approach Justification

The primary data collection technique employed in this study is web scraping, targeting the Institute of Electrical and Electronics Engineers (IEEE) website <https://ieeexplore.ieee.org> to gather reliable academic data, including author names, titles, abstracts, and publication metadata. The system's design explicitly addresses common user challenges, such as the time-consuming manual search for references and the cognitive complexity of synthesizing multiple sources.

The selection of the GPT 3.5 Turbo model is technically imperative and highly justified for this complex academic application. This model was chosen because it offers a significantly large context window capacity, which is essential for processing the substantial volume of input required for the user's topic along with the full text of multiple scraped abstracts necessary for meaningful academic synthesis. Furthermore, the model excels in instruction following and complex prompt engineering, capabilities that are nonnegotiable for automated academic writing. This proficiency allows the system to rigidly enforce structural constraints, implement specific citation styles, and ensure the generated output is semantically cohesive with the source abstracts. This efficiency and control directly validate its importance in this automated system.

Prior to this system's development, authors typically performed manual literature studies, a workflow illustrated in Figure 2. The system automates this process by leveraging web scraping from a trusted source (IEEE) and the highly controlled output of the GPT 3.5 Turbo model, thereby facilitating a substantial increase in user efficiency.

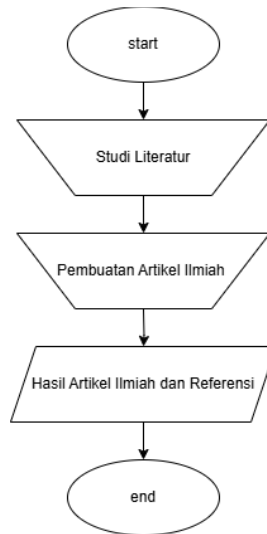


Fig. 2 Workflow System

### 3.2 Formal System Formulation

To rigorously define the automated article generation process, the system is formally conceptualized using a mathematical model that describes the synthesis function. This formulation clarifies the role of the inputs and constraints imposed on the language model via prompt engineering. The final scientific article  $A$  is defined as the output of the language model function  $\mathcal{G}$  (representing the GPT3.5 Turbo API call), which is strictly conditioned by the User Prompt  $P$  and the Reference Data  $R$  derived from the web scraping process, all while adhering to specific academic constraints  $(C_{struct}, C_{style})$ :

$$A = \mathcal{G}(P | R, C_{struct}, C_{style})$$

Where:

- $A$  is the Generated Scientific Article, structured according to the IMRaD format.
- $\mathcal{G}$  is the Text Generation Function implemented via the OpenAI API, which maximizes the likelihood of producing an academically appropriate token sequence.
- $P = \{Topics, Keywords, Format\}$  is the User-Defined Input parameters.
- $R = \{Abstract_1, Abstract_2, \dots, Abstract_3\}$  is the Scraped Reference Data (Abstracts and metadata) that serves as the factual basis for the article.
- $C_{struct}$  represents the Structural Constraints enforced through the prompt (e.g., mandatory IMRaD sections).
- $C_{style}$  represents the Stylistic Constraints (e.g., mandated citation format).
- The function  $\mathcal{G}$  fundamentally operates by maximizing the probability of a sequence of tokens  $w = w_1, w_2, w_3, \dots, w_k$  that forms the article, given the entirety of the input context  $I$ :

$$A = \frac{arg \max P(w | I)}{w}$$

### 3.3 System Workflow Process

The general system overview defines the core focus of the research, which is the development of an automated scientific article generator through a systematic design of the system and application. As shown in Figure 3, the system will implement web scraping processes and the OpenAI GPT language model service into the program code. The operating principle is to use a web application that serves as a bridge between the user via the User Interface (UI) and the OpenAI API service, while also leveraging web scraping for data retrieval. On the website's UI, users can input keywords related to their topic and

specify the number of references needed. This input is then used to perform the web scraping process. Subsequently, these keywords are used to construct a prompt for the GPT language model, which generates the article text that is then returned to the user on the website.

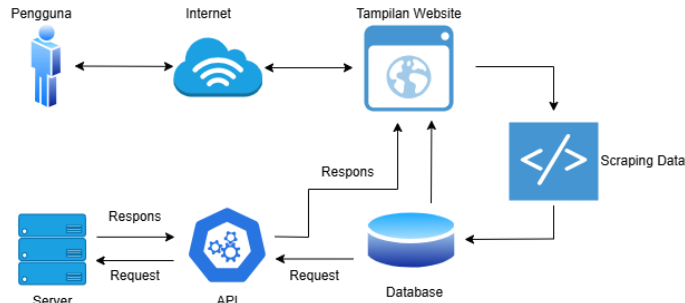


Fig. 3 General System Overview

The web scraping process is outlined in the flowchart in Figure 4. It begins with the user inputting a topic and the desired number of references. This request undergoes an input validation process; if it fails, the process stops, and the user must reenter the request. If successful, the process continues to the parsing stage. The system automatically opens the IEEE website, fills the search field with the provided topic, and initiates the search. It then searches for a specified number of articles by extracting the link for each one, which are then visited individually. This looping process continues by opening each article's detail page to retrieve information such as title, author names, abstract, publication date, DOI, URL, and publication name. The publication date data is then subjected to a pattern-matching technique (regular expression) to extract only the year of publication. The CSS selectors used for this process are detailed in Table 1.

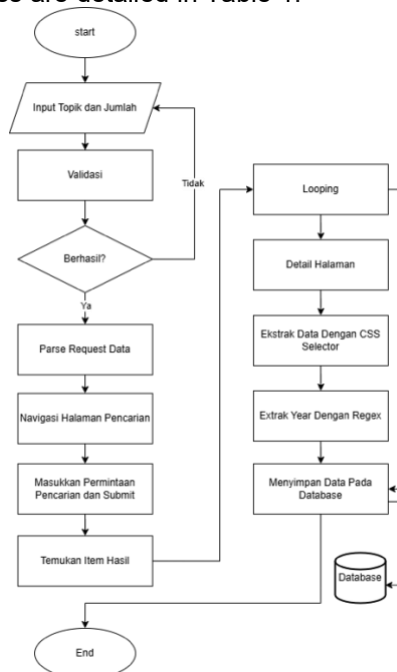


Fig. 4 Web Scraping Flowchart

Table. 1 CSS Selector

No.	DESCRIPTION	CSS Selector
1	Search Input	input.Typeahead-input
2	Article Link	result-item-title
3	Author Name	div.authors-info-container
4	Publication Year	div.doc-abstract-confdate
5	Detail Page Title	h1.document-title
6	Abstract	div.abstract-text
7	Publisher	div.stats-document-abstract-publishidIn a
8	DOI	div.stats-document-abstract-doi a

The article generation process, depicted in Figure 5, utilizes the OpenAI API's GPT 3.5 Turbo language model. The initial step involves the user inputting a topic and selecting a writing format, such as IEEE or APA. A validation check is performed on the user's input, and if the input is valid, it proceeds to the next stage of prompt construction. The prompt is structured in the code to include data retrieved from the database during the web scraping process. The OpenAI API is then called to generate the article based on the constructed prompt. The resulting article text is automatically displayed on the frontend, allowing the user to review and customize it as needed.

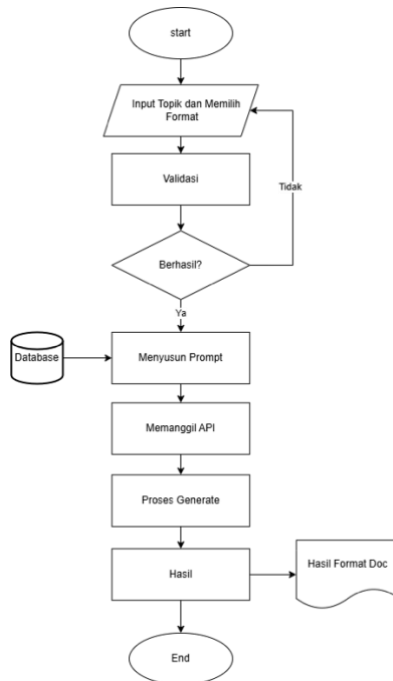


Fig. 5 Article Generation Process

The activity diagrams illustrate the various activities and processes within the system from start to finish. The diagram begins with the user accessing the web URL and submitting a request by selecting a language, inputting a topic, and choosing a format. A validation check ensures the request meets the criteria. Following validation, the system proceeds with scraping data from the IEEE website and saving it to the database. The user can then view, edit, or delete the retrieved data. This diagram shows the continuation of the process, connecting the reference search results with the OpenAI API. The user fills out a request form with the topic and chosen format. After validation, the user's request is used to create a text prompt that is sent to the GPT model to generate the article. The output is displayed directly on the UI and can be further customized by the user.

### 3.4 Evaluation Metrics

The proposed system underwent a rigorous testing phase aimed at assessing its feasibility and validity in generating scientific articles. This evaluation primarily utilized two quantitative methods: ROUGE and BERTScore, which measure the semantic and lexical congruence between the system generated text and reference documents.

#### 1. ROUGE (Recall Oriented Understudy for Gisting Evaluation)

ROUGE is a widely recognized set of metrics employed for evaluating automatically generated text, such as summaries or generated content, against a set of human-written reference texts [20]. It quantifies the similarity based on the  $n$ -gram overlap between the generated text and the reference. A higher ROUGE score indicates a greater degree of similarity between the generated text and the reference. In this study, ROUGE-1, ROUGE-2, and ROUGE-L were utilized as key evaluation metrics. ROUGE-1 assesses overlap based on single words (unigrams), ROUGE-2 measures overlap based on word pairs (bigrams), while ROUGE-L evaluates the similarity based on the Longest Common Subsequence (LCS) of words, reflecting sentence-level structure similarity.

#### Precision

Precision measures the proportion of words (or  $n$ -grams) in the system's output that are also present in the reference.

$$ROUGE - 1 \text{ Precision} = \frac{\text{Number of overlapping unigrams}}{\text{Total unigrams in system output}} \quad (1)$$

$$ROUGE - 2 \text{ Precision} = \frac{\text{Number of overlapping bigrams}}{\text{Total bigrams in system output}} \quad (2)$$

$$ROUGEL - L \text{ Precision} = \frac{LCS}{\text{Total word in system output}} \quad (3)$$

#### Recall

Recall measures the proportion of words (or  $n$ -grams) in the reference text that are successfully captured by the system's output.

$$ROUGE - 1 \text{ Recall} = \frac{\text{Number of overlapping unigrams}}{\text{Total unigrams in reference}} \quad (4)$$

$$ROUGE - 2 \text{ Recall} = \frac{\text{Number of overlapping bigrams}}{\text{Total bigrams in reference}} \quad (5)$$

$$ROUGEL - L \text{ Precision} = \frac{LCS}{\text{Total word in reference}}$$

#### F1-Score

The F1-Score represents the harmonic mean of Precision and Recall, providing a balanced measure that considers both false positives and false negatives.

$$F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

#### BERTScore

BERTScore is a metric designed to compute the semantic similarity between a reference sentence and a generated sentence. Unlike traditional overlap based metrics, BERTScore leverages contextual embeddings from the pretrained BERT model, enabling it to

understand word differences by considering their surrounding context. In its implementation, each word is transformed into an embedding, a numerical representation that encapsulates the word's meaning within its sentence context. These embeddings are then compared with the embeddings of the reference sentence using cosine similarity. BERTScore yields a score ranging from 0 to 1, where a value of 1 signifies perfect agreement between the two sentences, and 0 indicates complete dissimilarity. Consequently, a higher BERTScore indicates greater semantic correspondence. The equations for calculating BERTScore's Precision, Recall, and F1-Score are presented below:

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i^\top \hat{x}_j \quad (7)$$

Recall in BERTScore is obtained by matching tokens in the reference ( $x$ ) with tokens in the generated output  $\hat{x}$ .

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i^\top \hat{x}_j \quad (8)$$

The F1-Score for BERTScore combines precision and recall to provide a single, balanced metric.

$$F_{BERT} = 2 \cdot \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \quad (9)$$

## 4. Experimental Setup

The evaluation of the developed system was conducted through a controlled experiment focused on generating article drafts across various user defined topics. The primary data for analysis consisted of the generated article outputs, which were rigorously compared against a set of human written reference texts. These reference documents were precisely the abstracts of scholarly articles retrieved dynamically from the IEEE Xplore database during the system's initial web scraping phase.

For data analysis, the quality of the generated text was quantitatively evaluated using two established metrics: ROUGE and BERTScore. ROUGE was specifically utilized to measure lexical similarity based on n-gram overlap and the Longest Common Subsequence (LCS), thereby assessing the degree of shared words and phrases between the generated and reference texts. Conversely, BERTScore was employed to evaluate semantic similarity by leveraging contextual embeddings. This approach provided a deeper, context aware understanding of the conceptual alignment and meaning preservation between the two text sources. The final compiled results, including Precision, Recall, and F1-scores for both metrics, provided a comprehensive and multifaceted analysis of the system's performance feasibility and validity.

## 5. Result and Analysis

This section presents the primary findings derived from the implemented system and the application of the defined evaluation metrics. The results are reported objectively and logically, serving as the necessary foundation for the detailed interpretation provided in the subsequent discussion.

## 5.1. System Implementation Analysis

The developed system successfully integrates web scraping for data collection with the OpenAI API for sophisticated text generation. The backend is robustly constructed using the Django framework, complemented by an intuitive frontend developed with HTML, CSS, and JavaScript. The web scraping process is dynamically initiated by user input specifying the research topic and the desired number of articles. The system automates the scraping of essential metadata, including the title, authors, abstract, URL, year, DOI, and publication name. The user interface for the web scraping input is shown in Figure 6, while the final generated article is displayed via the interface shown in Figure 7.

Title	Authors	Year	Link	Action
Study on Detection Mechanism of HF Radar for Early Tsunami Detection and Comparison to Other Tsunami Sensors	Iswandi, Risanuri Hidayat, Budi Setiyanto, Sigit Budi Wibowo	2019	<a href="#">Link</a>	
Intelligent information retrieval for Tsunami detection using wireless sensor nodes	Deepali Virmani, Nikita Jain	2016	<a href="#">Link</a>	
Detection of tsunami wave generation and propagation using Fiber Bragg Grating sensors	A.S. Gurus Prasad, S. Asokan, Rao Tatavarti	2009	<a href="#">Link</a>	

Fig. 6 User Interface for The Web Scraping

The system demonstrated stable and responsive functional performance. Reference data acquired from the scraping process is persistently stored in the Django database, allowing for data reuse and verification. The Selenium library was specifically utilized for its capability to navigate complex search result pages and efficiently load dynamic content from the JavaScript heavy IEEE website. Crucially, functional tests confirmed the system's error free operation, from initial user input validation through to the final article generation.

**Development of Tsunami Detection Sensor Technology**

**Abstract**  
This article investigates the use of HF radar technology as an alternative method for detecting tsunamis, highlighting its advantages such as wide radar coverage, real-time capabilities, and continuous monitoring. The study compares HF radar technology with other tsunami sensors and concludes that HF radar can serve as an efficient tsunami sensor. However, further development in field testing and algorithm improvement is necessary to enhance detection accuracy.

**Introduction**  
Tsunamis pose a significant threat to coastal communities around the world, with devastating consequences for both human life and infrastructure. While traditional tsunami detection systems primarily rely on earthquake sensors to provide early warning alerts, recent events have highlighted the need for alternative technologies to enhance detection capabilities. In this context, High-Frequency (HF) radar technology has emerged as a promising solution for tsunami detection through the use of radio waves.

Total Words: 1239

Fig. 7 User Interface for the Generated Article

## 5.2. System Testing Analysis

The performance of the generated articles was quantitatively assessed using two primary metrics: ROUGE for lexical overlap and BERTScore for semantic similarity. The ROUGE test, which measures the lexical similarity between the generated and reference

texts, revealed key insights into the model's token-level behavior (results detailed in Figure 8). Specifically, the system achieved a high ROUGE-1 Recall value of 80.63%, indicating that a substantial proportion of the unigrams (single words) from the reference abstracts were successfully incorporated into the output. Conversely, the ROUGE-1 Precision was considerably lower at 28.32%. This disparity is interpreted as evidence that the model does not merely copy source text but actively introduces new words and phrases, demonstrating a capacity for original generation. The overall ROUGE-1 F1-Score reached 41.92%, significantly exceeding the ROUGE-2 (22.27%) and ROUGE-L (22.01%) scores. This result confirms the system's proficiency in matching individual terms but also highlights a limitation in replicating exact word sequences (bigrams) or overall reference sentence structures.

ROUGE Evaluation Results			
Method	Precision	Recall	F1-Score
ROUGE-1	28.32%	80.63%	41.92%
ROUGE-2	15.04%	42.89%	22.27%
ROUGE-L	14.87%	42.34%	22.01%

Fig. 8 Rouge Test Result

BERTScore Evaluation Results	
Method	Score (%)
Precision	83.01%
Recall	85.98%
F1-Score	84.46%

Fig. 9 BERTscore Test Result

In contrast, the BERTScore test, designed to assess deeper semantic similarity, yielded substantially higher results (Figure 9). The system achieved a Precision of 83.01% and a Recall of 85.98%. These high scores strongly suggest that, despite the structural differences noted by ROUGE, the generated articles are highly relevant in meaning and conceptual alignment to the source references. The robust F1-Score of 84.46% confirms the system's effectiveness in capturing the underlying semantic content. Furthermore, the analysis of the BERTScore per reference (presented in Figure 10) substantiates this finding, showing individual F1-Scores consistently ranging from 82.11% to 87.74%. This detailed breakdown verifies the system's capacity to synthesize information effectively from multiple, diverse sources without becoming reliant on a single dominant reference.

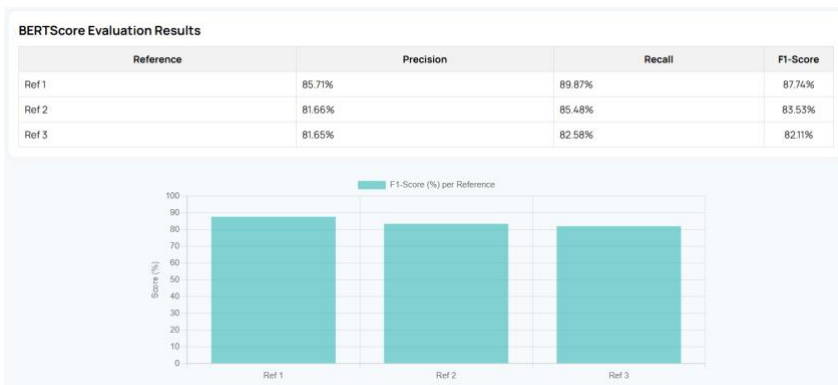


Fig. 10 BERTscore Test Result for each Reference

### 5.3 Discussion

The comparative analysis between the ROUGE and BERTScore results provides compelling insights into the system's synthesis capability. The high ROUGE-1 Recall of 80.63% is a significant technical finding, proving the system's efficiency in identifying and incorporating relevant key terms and phrases from the reference abstracts. The concurrent low ROUGE-1 Precision of 28.32% is crucial, as it suggests that the model is performing true paraphrasing and original generation rather than simple text extraction.

The most profound interpretation, however, is supported by the remarkably high BERTScore F1-Score of 84.46%. This score is the central finding of this study. It conclusively demonstrates that despite the lower ROUGE-L scores indicating divergence in sentence structure, the generated text is semantically equivalent to the original references. This highlights the system's core strength: its ability to comprehend the conceptual essence of multiple sources and synthesize it into new, original sentences that successfully preserve the intended meaning. Finally, the consistent per-reference BERTScore analysis (Figure 10) unequivocally confirms that the system effectively processes and synthesizes information from all provided scholarly references, establishing its validity as a comprehensive literature synthesis tool.

## 6. Conclusion

This study successfully developed an automated scientific article generator system that leverages the integration of dynamic Web Scraping using GPT algorithm. The system effectively addresses the long-standing academic challenge of manually synthesizing information from multiple scholarly sources to produce new, cohesive texts, thereby significantly reducing the time required for literature synthesis and initial drafting. The quantitative evaluation, employing ROUGE (for lexical overlap) and BERTScore (for semantic similarity), confirmed the system's effectiveness and revealed its unique capabilities. The high ROUGE-1 Recall of 80.63% demonstrated strong proficiency in capturing essential vocabulary and key terms from the reference abstracts. Crucially, the lower ROUGE Precision indicated that the underlying language model actively generates original content rather than merely replicating the source material. The most significant finding is the high BERTScore F1-Score of 84.46%. This result conclusively proves the system's core strength: its ability to comprehend and successfully rephrase the underlying semantic meaning of complex references. This system not only capable of generating structured text but also excels at synthesizing conceptual information.

As future work, research can be directed toward several key areas to further enhance the system's capabilities. Firstly, integrating more advanced Large Language Models, such as GPT-4 or specialized fine-tuned models, could potentially improve the semantic quality and structural complexity of the generated articles. Secondly, expanding the reference collection mechanism to include other major academic databases (e.g., Scopus or Google Scholar) would increase the breadth and diversity of source material. Finally, incorporating a post-generation self-correction module that utilizes specialized metrics to evaluate the academic rigor of the generated output before final display could further optimize the system's overall performance and reliability.

## References

- [1] Yanti, E. Salsabila, N. Khoirun Nisa, A. Rapindo, S. Anggraini, and E. Tamara, "Scientific Work," 2024, [Online]. Available: <https://jicnusantara.com/index.php/jiic>
- [2] Akhmad and D. Jurusan Administrasi Niaga Politeknik Negeri Ujung Pandang, *Proceedings of the Research Results Seminar (SNP2M)*.
- [3] R. Agustina and A. Ikhlas, "Analysis of Student Difficulties in Writing Scientific Papers at STKIP Muhammadiyah Sungai Penuh," *Pendekar: Character Education Journal*, vol. 5, p. 6, Sep. 2022, doi: 10.31764/pendekar.v5i4.11565.
- [4] Widodo, A. Kadir Jailani, S. Novitasari, D. Sutisna, and M. Erfan, "Analysis of the Paper Writing Ability of New PGSD Students at the University of Mataram."
- [5] S. Satriajati, S. Panuntun, and S. Pramana, "Implementation of Web Scraping in the Collection of Criminal News During the Covid-19 Pandemic," *National Seminar on Official Statistics*, vol. 2020, no. 1, pp. 300–308, Jan. 2021.
- [6] P. Alia, J. Prayogo, R. Kriswibowo, and A. Setyadi, "Implementation of Open Artificial Intelligence ChatGPT Integrated with WhatsApp Bot," *Advance Sustainable Science, Engineering and Technology*, vol. 6, p. 2401019, Sep. 2024
- [7] Z. Wahyuzi and Abdurrahim, "Integrating Causal Analysis with Machine Learning for Meteorological-Based Energy Consumption Forecasting: A Multi-Method Approach," *International Journal of Informatics and Computation*, vol. 7, no. 2, pp. 384–399, Jul. 2025.
- [8] Y. Yuniati, K. M. Fitria, Melvi, S. Purwiyanti, E. Nasrullah, and M. A. Muhammad, "Performance Analysis of GPT-3 Content Extraction Using BERTScore and ROUGE Metrics," *Journal of Information Technology and Computer Science*, vol. 11, no. 6, pp. 1273–1280, Dec. 2024, doi: 10.25126/jtiik.2024118088.
- [9] H. Kristianto et al., "Optimization of E-Commerce Product Descriptions: Utilizing AI ChatGPT for Engaging and Contextual Descriptions," *Journal of Informatics Engineering and Information Systems*, vol. 11, no. 3, 2024, [Online]. Available: <http://jurnal.mdp.ac.id>
- [10] S. Rahayu, J. L. Wattimena, and the Ambon Religious Training Center, "Utilization of Artificial Intelligence (AI) in Scientific Article Writing."
- [11] Z. Munawar, H. Soerjono, N. I. Putri, Hernawati, and A. Dwijayanti, "The Benefits of Artificial Intelligence ChatGPT for Assisting Scientific Writing," *TEMATIK*, vol. 10, no. 1, pp. 54–60, Jun. 2023, doi: 10.38204/tematik.v10i1.1291.
- [12] G. N. A. W. S. Putra, I. G. J. E. Putra, and I. N. Purnama, "Leveraging Energy Efficiency of Smart Room Using Blink System," *International Journal of Informatics and Computation*, vol. 7, no. 2, pp. 374–383, Jul. 2025, doi: 10.35842/ijicom.v7i2.127.
- [13] Vaswani, N. Shazeer, N. Parmar, and J. Uszkoreit, "Attention Is All You Need," *ArXiv*, Jun. 2017, Accessed: Sep. 04, 2025. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [14] M. L. Hidayat and K. D. Irianto, "AutoMedic: Framework of Automatic Pill Dispenser System with Human-Centered Design Method," *International Journal of Informatics and Computation*, vol. 7, no. 2, pp. 400–418, Aug. 2025, doi: 10.35842/ijicom.v7i2.136.
- [15] R. OpenAI, K. N. OpenAI, T. S. OpenAI, and I. S. OpenAI, "Improving Language Understanding by Generative Pre-Training." [Online]. Available: <https://qluebenchmark.com/leaderboard>
- [16] T. Zhang, V. Kishore, F. Wu, K. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," Sep. 2019, doi: 10.48550/arXiv.1904.09675.
- [17] S. D. Saputra, I. Syahrohim, V. H. Pranatawijaya, N. Noor, and K. Sari, "Implementation of a Web-Based ChatGPT Summarizer API," *Journal of Informatics and Applied Electrical Engineering*, vol. 12, no. 3, pp. 2830–7062, doi: 10.23960/jitet.v12i3.4313.
- [18] R. Aryabimo, D. Bernady, N. N. K. Sari, and V. H. Pranatawijaya, "Implementation of the ChatGPT API in a Web-Based Restaurant Application," *Journal of Informatics and Applied Electrical Engineering*, vol. 12, no. 3, Aug. 2024, doi: 10.23960/jitet.v12i3.4408.
- [19] E. Mu'alifah, S. Farisa, and C. Haviana, "Design of a Scientific Article Summarization System Using GROBID and Large Language Models (LLM) Based on SciBERT."
- [20] Z. Idhafi, S. Agustian, F. Yanto, and N. S. Hidayat, "Automatic Text Summarization in Indonesian-Language Articles Using the Maximum Marginal Relevance Method," *Journal of Computer Science and Information Technology (CoSciTech)*, vol. 4, no. 3, pp. 609–618, Dec. 2023, doi: 10.37859/coscitech.v4i3.6311.