Design of text generator application with OpenAI GPT-3

Kaira Milani Fitria[1]
[1]Faculty of Computer Science, Informatics & Business Institute Darmajaya, Lampung, Indonesia

Article Info

Article history:

Received 16 Juni 2023
Revised 20 Juni 2023
Accepted 27 Oktober 2023

Keywords:

Artificial Intelligence
GPT-3
NLP
Text Generation
Web Application
First keyword

ABSTRACT

The increasing need for text content creation today challenges the development of systems that can alleviate the need for text creation. Currently, text generation is done manually and has various shortcomings, especially in terms of time constraints, human error, limited creativity, and writing that tends to be repetitive by certain people, which can cause a decrease in quality and diversity in the sentences produced. This research was conducted by designing an AI-based text generator application using the GPT-3 language model to generate text automatically and help overcome some obstacles. Applying this app will increase efficiency and productivity, increase the writer's ideas and creativity, automate routine tasks, and produce exciting and communicative sentences. The app's ability to generate text quickly and accurately and be personalized makes it valuable in various fields. The method used in this research is implementing the GPT-3 language model APIs into the text generator application created so that the application can connect with the GPT-3 engine that has been modified in its prompting method. The output of this application is a text according to the context needed by the user through keywords previously inputted on the web interface. The result of this research is creating a text generator application that implements the GPT-3 language model API that can run well during running tests. And the testing results with the black box testing method provide valid results related to the web system created. It means the application can work well according to its function, which is to display text results that match the context inputted by the user.

Corresponding Author:
Kaira Milani Fitria,
Faculty of Computer Science, Department of Informatics Engineering
Informatics & Business Institute Darmajaya
Jl. ZA. Pagar Alam No.93, Gedong Meneng, Bandar Lampung, Lampung, Indonesia 35141
Email : kairaamilanii@gmail.com

1. INTRODUCTION

AI is a broad notion that includes a variety of technologies, such as machine learning, deep learning, computer vision, natural language processing (NLP), robots, and ML (machine learning). These technologies enable systems to learn from data without explicit programming [1]. Artificial intelligence (AI) has become commonplace in our daily lives and is rapidly expanding [2]. NLP is utilized in various applications, including sentiment analysis in investment, chatbots in customer relationship management, decision support systems in retail and operations, and legal, architectural, and transportation [3]. NLP has numerous clinical uses, such as aiding with provider documentation, automated structured chart abstraction, and machine learning [4]. Natural language processing (NLP) is a field that aims to enable people to achieve this. NLP refers to the automatic computational processing of human languages as a whole. It consists of both algorithms that accept text created by humans as inputs and algorithms that generate text with a natural appearance as outputs [5]. The work of language modeling serves as the foundation for model pre-training in natural language processing (NLP). Given a past of unannotated texts, language modeling aims to predict the following token [6]. The training paradigm for language modeling has started to be dominated by deep learning techniques. Recurrent neural networks (RNNs) were frequently employed in early techniques for neural language modeling [7], [8].Long short-term memory (LSTM) stands out among the RNN family because it has the benefit of being less susceptible to the gradient vanishing problem through its well-designed deep learning methods, which have started to rule the training paradigm of language modeling.

Recurrent neural networks (RNNs) were frequently employed in early techniques for neural language modeling. Long short-term memory (LSTM) stands out within the RNN family due to its benefit of being less vulnerable to the gradient vanishing problem due to its well-designed architecture [9]. The "pre-training then fine-tuning" paradigm has since begun to take shape. The first model to use unidirectional transformers as the framework for the GPT of language models was generative pre-training (GPT), demonstrating the striking potential of pre-training techniques for various downstream applications [10]. While there are undoubtedly many benefits, it has also created new, challenging problems that must be solved [11]. To address these issues and move further, it is essential to understand some key concepts related to AI and GPT (Generative Pre-Trained Transformer). An AI model called the Generative Pre-Trained Transformer-3 (GPT-3) learned unsupervised on a significant corpus of text data. GPT-3 can now make responses in written language that mirror those produced by humans thanks to this training [12]. This year's major AI news was the mid-year release of OpenAI's GPT-3 language model, which can produce natural language documents that can be remarkably difficult to discern from stuff written by humans. Much more than people often anticipate technical development in NLP, it gained coverage in both the technical and general media [13]. These robust neural network-based models can be utilized for various NLP tasks, and because of how complex their language-generating capabilities have evolved, it can be quite challenging to tell their outputs apart from human language [14].

As a "task-agnostic" language model, Pre-trained Transformer 3 (GPT-3) may enable further NLP applications. In many use situations and on many benchmarks, the system, which the research facility OpenAI published for testing in August 2020, can compete with older, fine-tuned models without any more fine-tuning. Its developers claim that it was pre-trained using a dataset more than ten times bigger than any used by comparable models, including Wikipedia and a filtered version of the Common Crawl dataset [1]. OpenAI created the powerful huge language model GPT-3. Through knowledge distillation, its knowledge can be applied to jobs farther down the line [15]. More substantial natural language processing (NLP) and machine learning (ML) capabilities enable more powerful models to perform with better understanding and response abilities rather than being designed for performing human communication interactions [16]. Users can access GPT-3 and "program" it by providing words in human language, instructions on what to do, and, optionally, a few samples of the intended output through an application programming interface (API) that OpenAI has built up. In contrast to prior NLP models, which require massive volumes of training data, GPT-3 receives a comparable number of instructions to those humans would receive when asked to do a task. This gives it the flexibility required for a variety of applications. GPT-3 is a versatile language model that can be used for various activities. GPT-3 has demonstrated its ability to produce writing that can be mistaken for a human author, including poetry, articles, web interface code, and descriptions of products or jobs [13].

The Generative Pre-trained Transformer 3, or GPT-3, was created by OpenAI, employing computer-based analysis of vast volumes of textual (natural language) data made available to the public [1]. GPT-3 and other technologies can nearly pass the Turing Test, determining whether these programs' language can be easily distinguished from human-produced language [17], [18].GPT-3 functions with a small amount of fine-tuning data. The language model knows practically every domain by nature. Only a few instructions on what to do and, preferably, a few instances of the desired result are used to teach it (few-shot learning) [19]. The "text-davinci-003" model now in use makes use of a training dataset made up of 45 million web pages, books, and other sources [20]. As previously mentioned by other authors, GPT-3 demonstrated excellent cooperation and produced pertinent, accurate, and proper research [21], [22]. The successor to GPT-2, GPT-3, increased the data scale (45 TB vs. 40 GB) and the parameter space (175 billion vs. 1.5 billion), making it the largest language model ever made. The model performs outstandingly in zero-shot and few-shot conditions and can complete downstream tasks without fine-tuning [23]. GPT-3 has shown good results on various new tasks, including arithmetic addition, news article generation, language interpretation, and code writing, based on the multi-task generalization skills of GPT-2. This model will be increasingly more powerful when the number of parameters increases.

AI models can be connected to platforms like websites, messaging services, and mobile apps via their APIs (Application Programming Interfaces) and utilized for various functions like customer support, information retrieval, and personal help [20]. The ability to annotate data for machine learning model training without the need for specific in-domain knowledge is one way that GPT-3 may be of assistance to people [24]. Prompt-Learning, sometimes referred to as Prompting, provides insight into what natural language processing (NLP) may include in the future [25]. Prompting techniques can be integrated as input in the prompt generation to the GPT-3 engine accessed through the language model API. This method is the basic technique in making the text generator application in this research.

2. METHOD

2.1 Research Framework

A set order of frameworks in the research methodology must be followed. This order of frameworks serves as an example of the procedures that must be taken to ensure the success of this research. The following structure is :

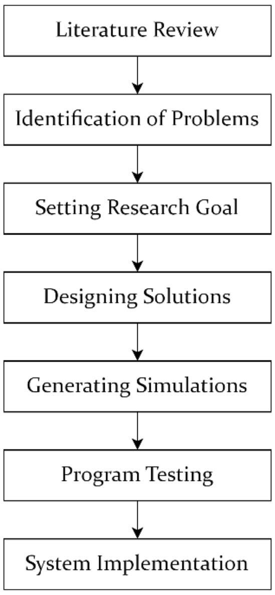


Figure. 1. Research Framework

2.2 Research Framework Description

Based on Figure 1, the explanation for each step in the research framework cas be described as follows:

a. Literature Review

A literature review is a critical and systematic analysis of published research and scholarly articles relevant to a particular topic or research question. The primary purpose of a literature review is to establish the current state of knowledge on a given topic, identify gaps, conflicts, or inconsistencies in the existing literature, and determine the need for further research. Some of the primary references that are referred to in this research are Vaswani [26] in the study "Attention is All You Need, Advances in neural information processing systems", J. Devlin [27] in the study "BERT: Pre-training of deep bidirectional Transformers for language understanding", Radford [28] in "Language Models are Unsupervised Multitask Learners", T. B. Brown [1] in "Language models are few-shot learners".

b. Identification of Problems

The identification of problems in this study starts from the rise of the growing e-commerce sites, raising an issue, namely competition in the marketing process. A digital business must require copywriting skills. The solution to improve copywriting skills for ordinary people is by making an application that can provide exciting sentence ideas that make it easier for copywriters to compose sentences by utilizing Artificial Intelligence (AI).

c. Setting Research Goal

The focus was determined based on the results of problem identification and the motivation that prompted the research. This research focuses on creating a text generator application that can be accessed publicly. The application design also contains various supporting features such as product/service descriptions, job descriptions, social media captions, email marketing templates, advertisements, and business promotion ideas. Features are also built to generate good reports for promotional videos without leaving the need for more attractive marketing methods, youtube video title ideas, word and sentence improvements, paragraph summarization, and keyword generation from a description, so content is included in global searches to increase promotional content insight.

d. Designing Solutions

The design of solutions based on the research focus is done with the copywriting application development method in conducting system design and application development. Implementing the GPT-3 language model in the application is by training on the language model and using the API code or service from this language model in the web program code that has been made before.

e. Generating Simulations

Based on the design of the solution created, a demonstration was built to test the application and started to see the suitability of the design with the expectations to be achieved. Demonstrating an app before its release serves several essential purposes. "Generating simulations" in creating an app refers to developing and implementing simulated environments or scenarios within the application. Generating simulations for an app involves designing the virtual environment, creating and implementing the necessary algorithms and models, integrating user interactions, and ensuring the simulation behaves realistically or as intended. This often requires computer graphics, physics modeling, artificial intelligence, and user interface design expertise. Overall, generating simulations in an app adds a dynamic and interactive element, allowing users to engage with virtual environments, scenarios, or systems that closely resemble real-world experiences. It enhances user engagement, facilitates learning, enables experimentation, and provides useful functionality depending on the specific goals and objectives of the app.

f. Program Testing

Program testing involves systematically evaluating an application to identify defects, errors, or issues and ensuring it functions as intended. The primary objective of program testing is to improve the app's quality, reliability, and usability before it is released to users. This plan outlines the testing objectives, scope, scenarios, cases, and resources required for testing. It helps ensure that testing activities are well-organized and cover all critical aspects of the app. User Acceptance Testing involves testing the app with end-users or stakeholders to ensure it meets their requirements and expectations. It often takes place in a real-world environment to simulate actual usage scenarios.

g. System Implementation

System implementation involves deploying and integrating the developed software system into a live operational environment. It encompasses a series of activities that transition the app from the development stage to its active use by end-users. The system implementation phase is crucial as it marks the transition from development to actual app usage. Successful implementation involves careful planning, proper infrastructure setup, rigorous testing, user training, and a smooth transition to the live environment. It sets the foundation for the app's long-term operation and lays the groundwork for subsequent maintenance and improvement activities.

2.3 System Overview

This system focuses on designing a web application that is made to be a link between the language model API service and the user through the User Interface (UI), where the website display will receive input from the user, then take the input and serve as a prompt for the GPT-3 language model service which then the results of the language model will be received by the web and displayed back to the web display visible to the user.

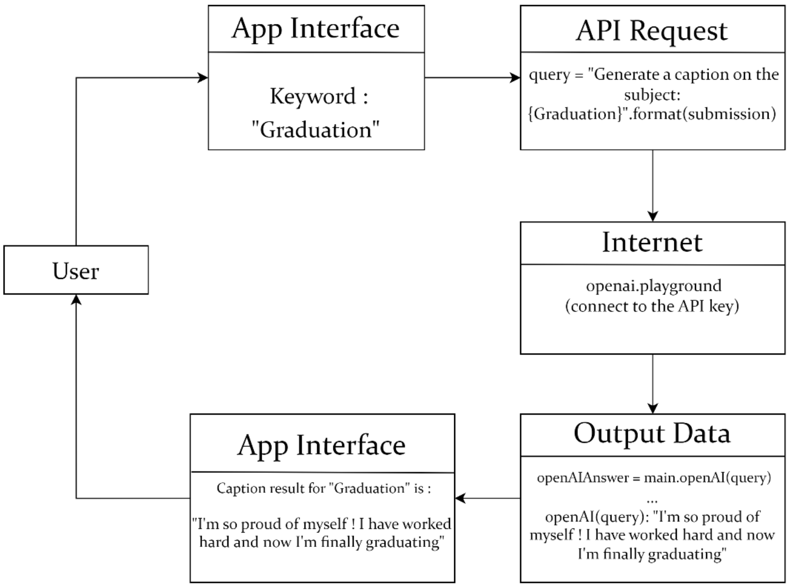

App Interface
Keyword : "Graduation"
API Request
query = "Generate a caption on the subject: {Graduation}".format(submission)
Internet
openai.playground
(connect to the API key)
Output Data
openAIAnswer = main.openAI(query)
...
openAI(query): "I'm so proud of myself ! I have worked hard and now I'm finally graduating"
App Interface
Caption result for "Graduation" is :
"I'm so proud of myself ! I have worked hard and now I'm finally graduating"
User

Figure. 2. System Overview

The application-building component created uses API from OpenAI GPT-3, which is embedded into the program code. API code provided by OpenAI through the author's developer account. Making applications is done with Visual Studio Code software with references from various sources such as GitHub, StackOverFlow, Youtube, OpenAI, arXiv, Papers with Code, and others. The program code used is Python 3.8, using several supporting libraries such as Flask and OpenAI libraries such as Flask and OpenAI for web display (frontend) using HTML, CSS, and JavaScript programming languages. In making the website layout design, the Adobe Illustrator application is used to make the details of the graphic components, the application to create complex visual elements, and the preparation of prototype website layout designs using the Figma application. Application testing application before deployment is carried out on localhost by running the program script on the Command Prompt or Visual Studio Code that opens in the Chrome browser. Chrome browser. After the application has run well on localhost, the program folder is deployed to the PythonAnywhere website so that the website has a URL that various devices can access.

2.4 System Activity Design

Activity diagrams are created to model visually the application design system developed by uniting some information in the software architecture. The type of UML diagram that will be applied in the design of this application is an activity diagram. The UML diagram in Figure.3 describes the modeling of how the application works from the user and system side.

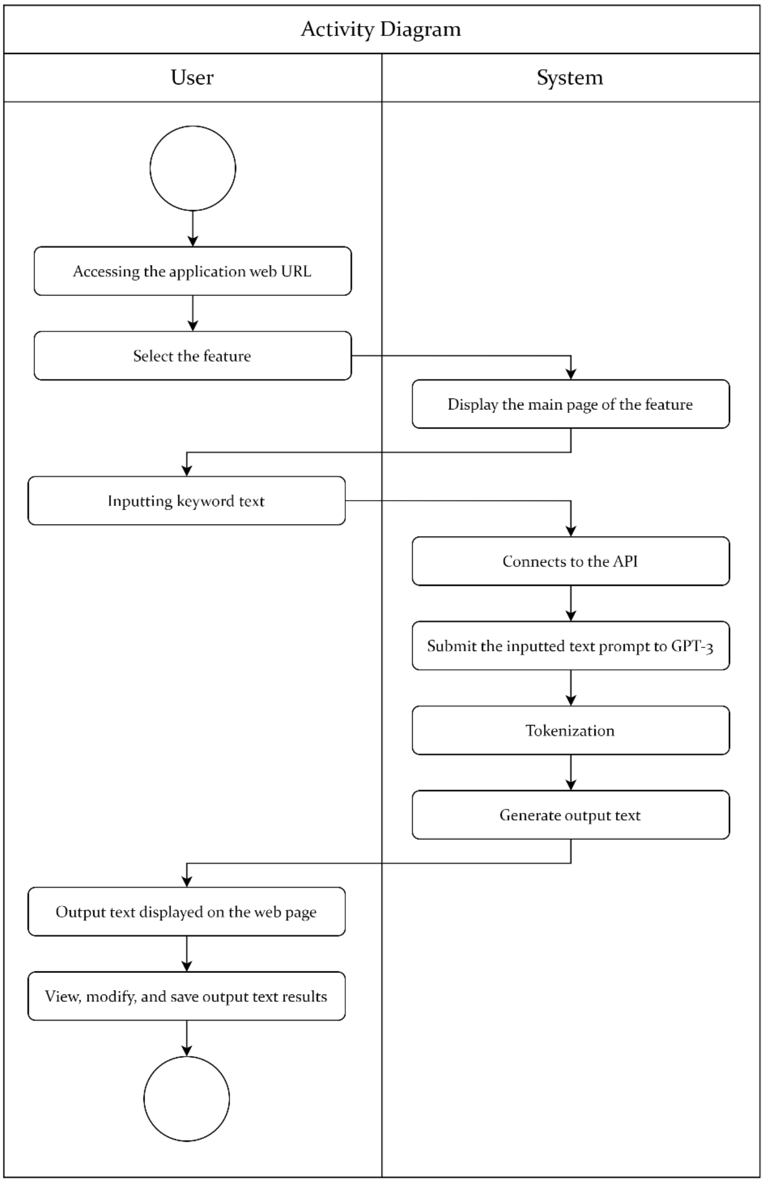
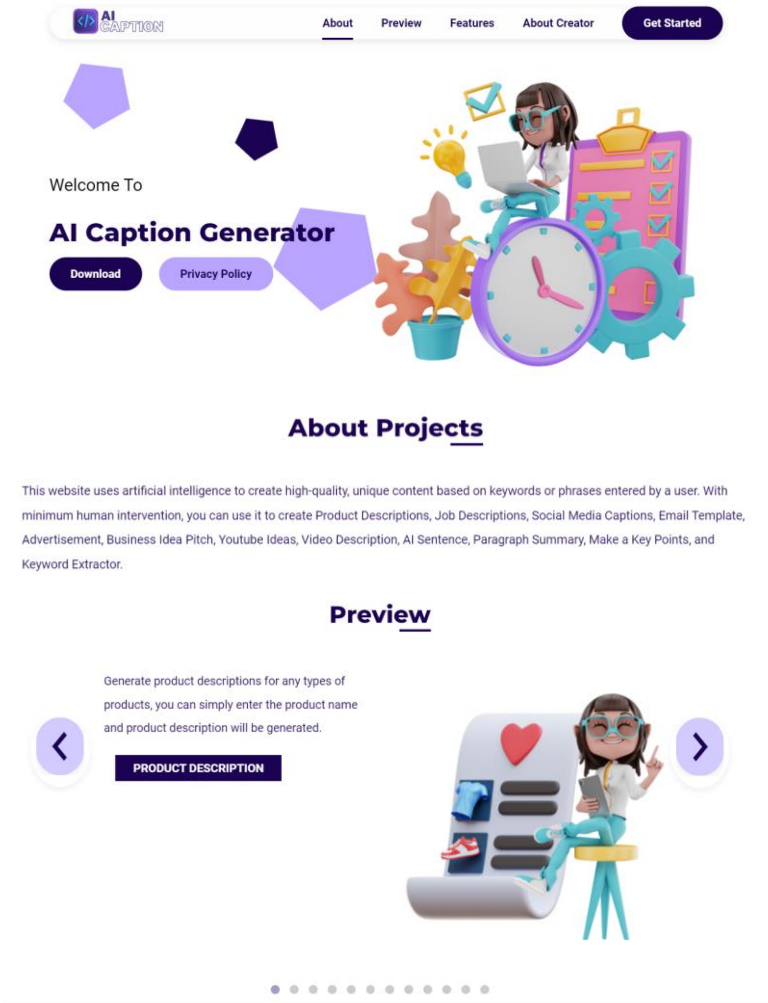

Activity Diagram
User
System
Accessing the application web URL
Select the feature
Display the main page of the feature
Inputting keyword text
Connects to the API
Submit the inputted text prompt to GPT-3
Tokenization
Generate output text
Output text displayed on the web page
View, modify, and save output text results

Figure. 3. System Activity Design

3. RESULTS AND DISCUSSION (10 PT)

3.1. Application Web Interface

The result of the website deployment process is forming a Uniform Resource Locator (URL) address as the website address. (URL) as the website address. Anyone who is connected to the internet network can access the website address that is already available. The web interface is shown in Figure 4.



This website uses artificial intelligence to create high-quality, unique content based on keywords or phrases entered by a user. With minimum human intervention, you can use it to create Product Descriptions, Job Descriptions, Social Media Captions, Email Template, Advertisement, Business Idea Pitch, Youtube Ideas, Video Description, AI Sentence, Paragraph Summary, Make a Key Points, and Keyword Extractor.

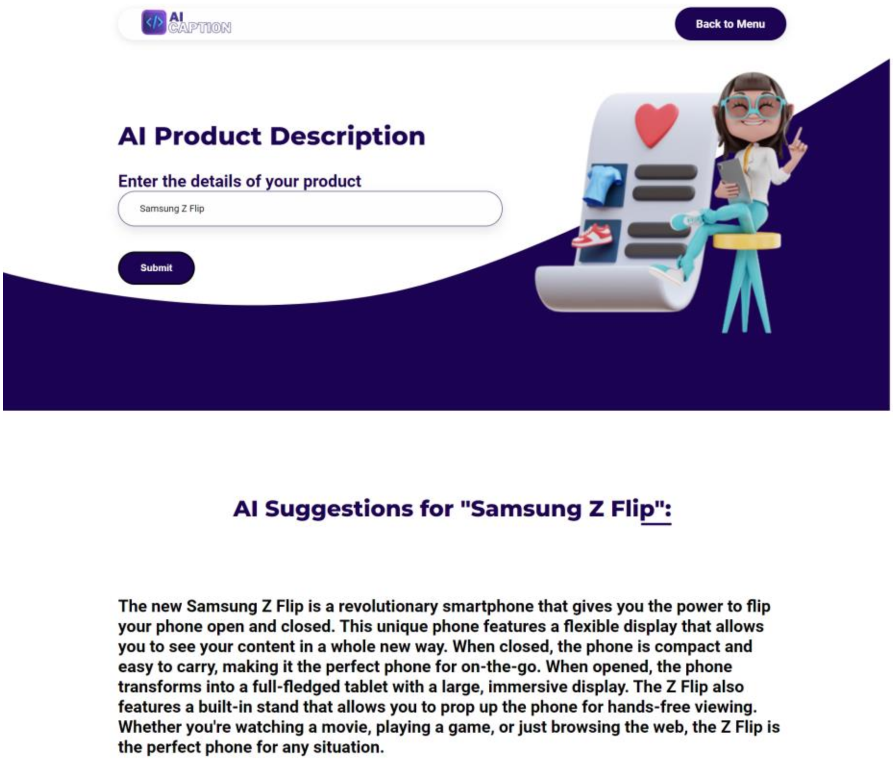
Generate product descriptions for any types of products, you can simply enter the product name and product description will be generated.

Figure. 4. Web app landing page

The explanation for several segments in the AI Caption Generator web starts from the "About Projects" part, which explains the purpose of why this application was created. The "Preview" segment is made with the aim that users who access the web will get an explanation of the function of each available feature according to its role. For example, one of the features is "Product Description" which has a description "Generate product descriptions for any type of products, you can simply enter the product name, and the product description will be generated".

3.2. Performance Result

Testing on features accessed on the AI Caption Generator web will display text results according to the inputted context. The tests include the success of the system in making API requests and displaying the output of the GPT-3 language model, the arrangement of the web layout, the responsiveness of the web display, and navigation to other pages. The results of the test running application when making calls from the GPT-3 API were successful, as shown in Figure 5.



AI Suggestions for "Samsung Z Flip":

The new Samsung Z Flip is a revolutionary smartphone that gives you the power to flip your phone open and closed. This unique phone features a flexible display that allows you to see your content in a whole new way. When closed, the phone is compact and easy to carry, making it the perfect phone for on-the-go. When opened, the phone transforms into a full-fledged tablet with a large, immersive display. The Z Flip also features a built-in stand that allows you to prop up the phone for hands-free viewing. Whether you're watching a movie, playing a game, or just browsing the web, the Z Flip is the perfect phone for any situation.

Figure. 5. Result of text generator app

3.3 Testing

Blackbox testing ensures that the entire system is tested and runs well, as users and developers expect. The stages carried out during blackbox testing are testing the system by accessing the web application through a web browser, testing text input and application output results, and testing the button functions found on each application page whether they function correctly. The results of blackbox testing on the text generator application can be seen in Table 1.

Table 1. Blackbox Testing Result

Action	Expected Result	Validity
Click the button on Preview	Move to the next menu along with explanation	Valid
Access the Features menu	Redirects to the section page about the choice of several features that can be accessed	Valid
Click the features button	Redirects to the features page	Valid
Access the About Creator menu	Redirects to a section page that contains web creator information	Valid
Access the Privacy Policy menu	Redirects to a section page about the web privacy rules	Valid
Input prompt text	The inputted text can be seen in the text box on the web page	Valid
Click the Submit menu	Display the output text	Valid
Click the web icon logo	Return to the main page	Valid
Click back to menu	Return to the main page	Valid

Based on Table 1, the testing results with the blackbox testing method provide valid results related to the web system created. The web application developed fully has functionality by the available features, and all features can be operated by the user because there is already an explanation in the web display.

4. CONCLUSION

Based on the results of making a text generator application with the implementation of the GPT-3 API, the application can run well. The application runs smoothly on localhost testing and runs well when accessed via a web server. The application can also work by displaying the results of several texts that match the context inputted by the user. The hope is that this application can be further developed by applying more prompting techniques so that the features in the application can be more extensive and varied.

REFERENCES

[1] T. Brown et al., “Language Models are Few-Shot Learners,” in Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[2] Z. Latinovic and S. C. Chatterjee, “Achieving the promise of AI and ML in delivering economic and relational customer value in B2B,” J Bus Res, vol. 144, pp. 966–974, 2022, doi: https://doi.org/10.1016/j.jbusres.2022.01.052.

[3] K. R. Chowdhary, “Natural Language Processing,” in Fundamentals of Artificial Intelligence, K. R. Chowdhary, Ed., New Delhi: Springer India, 2020, pp. 603–649. doi: 10.1007/978-81-322-3972-7_19.

[4] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, “Natural language processing: an introduction,” Journal of the American Medical Informatics Association, vol. 18, no. 5, pp. 544–551, Sep. 2011, doi: 10.1136/amiajnl-2011-000464.

[5] M. Bahja, “Natural Language Processing Applications in Business,” in E-Business, R. M. X. Wu and M. Mircea, Eds., Rijeka: IntechOpen, 2020, p. Ch. 4. doi: 10.5772/intechopen.92203.

[6] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Comput Speech Lang, vol. 13, no. 4, pp. 359–394, 1999, doi: https://doi.org/10.1006/csla.1999.0128.

[7] M. Sundermeyer, R. Schlüter, and H. Ney, LSTM Neural Networks for Language Modeling. 2012. doi: 10.21437/Interspeech.2012-65.

[8] T. Mikolov and G. Zweig, “Context dependent recurrent neural network language model,” in 2012 IEEE Spoken Language Technology Workshop (SLT), 2012, pp. 234–239. doi: 10.1109/SLT.2012.6424228.

[9] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[10] A. Radford, K. Narashiman, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-Training,” OpenAI, 2018, [Online]. Available: https://openai.com/blog/language-unsupervised/

[11] C. M. Gevaert, M. Carman, B. Rosman, Y. Georgiadou, and R. Soden, “Fairness and accountability of AI in disaster risk management: Opportunities and challenges,” Patterns, vol. 2, no. 11, p. 100363, 2021, doi: https://doi.org/10.1016/j.patter.2021.100363.

[12] A. Chan, “GPT-3 and InstructGPT: technological dystopianism, utopianism, and ‘Contextual’ perspectives in AI ethics and industry,” AI and Ethics, vol. 3, no. 1, pp. 53–64, 2023, doi: 10.1007/s43681-022-00148-6.

[13] R. Dale, “GPT-3: What’s it good for?,” Nat Lang Eng, vol. 27, no. 1, pp. 113–118, 2021, doi: DOI: 10.1017/S1351324920000601.

[14] D. Luitse and W. Denkena, “The great Transformer: Examining the role of large language models in the political economy of AI,” Big Data Soc, vol. 8, no. 2, p. 20539517211047736, Jul. 2021, doi: 10.1177/20539517211047734.

[15] S. Y. Kim, H. Park, K. Shin, and K.-M. Kim, “Ask Me What You Need: Product Retrieval using Knowledge from GPT-3,” Jul. 2022, [Online]. Available: http://arxiv.org/abs/2207.02516

[16] G. Generative Pretrained Transformer, A. Osmanovic Thunström, S. Steingrimsson, and S. Steingrimsson Can, “Can GPT-3 write an academic paper on itself, with minimal human input? GPT-3 write an academic paper on itself, with minimal human input?” [Online]. Available: www.openai.com

[17] A. M. TURING, “I.—COMPUTING MACHINERY AND INTELLIGENCE,” Mind, vol. LIX, no. 236, pp. 433–460, Oct. 1950, doi: 10.1093/mind/LIX.236.433.

[18] Kevin Lacker, “Giving GPT-3 a Turing Test,” Jul. 06, 2020.

[19] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a Few Examples: A Survey on Few-Shot Learning,” ACM Comput. Surv., vol. 53, no. 3, Jun. 2020, doi: 10.1145/3386252.

[20] D. Haluza and D. Jungwirth, “Artificial Intelligence and Ten Societal Megatrends: An Exploratory Study Using GPT-3,” Systems, vol. 11, no. 3, 2023, doi: 10.3390/systems11030120.

[21] R. Singh, V. Garg, and GPT-3, “Human Factors in NDE 4.0 Development Decisions,” J Nondestr Eval, vol. 40, no. 3, p. 71, 2021, doi: 10.1007/s10921-021-00808-3.

[22] Thomas J Ackermann, “GPT-3: a robot wrote this entire article. Are you scared yet, human?,” Artificial Intelligence: ANI, LogicGate Computing, AGI, ASI, Nov. 29, 2020.

[23] M. Zhang and J. Li, “A commentary of GPT-3 in MIT Technology Review 2021,” Fundamental Research, vol. 1, no. 6, pp. 831–833, 2021, doi: https://doi.org/10.1016/j.fmre.2021.11.011.

[24] B. Ding, C. Qin, L. Liu, L. Bing, S. Joty, and B. Li, “Is GPT-3 a Good Data Annotator?,” Dec. 2022, [Online]. Available: http://arxiv.org/abs/2212.10450

[25] B. Lester, R. Al-Rfou, and N. Constant, The Power of Scale for Parameter-Efficient Prompt Tuning. 2021. doi: 10.18653/v1/2021.emnlp-main.243.

[26] A. Vaswani et al., “Attention is all you need,” Adv Neural Inf Process Syst, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[27] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1, no. Mlm, pp. 4171–4186, 2019.

[28] I. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, “Language Models are Unsupervised Multitask Learners,” OpenAI Blog, 2019, [Online]. Available: http://arxiv.org/abs/2007.07582