https://doi.org/10.15407/csc.2024.03.068
UDC 004.942

A.M. SHVYNDIA, Student, Taras Shevchenko National University of Kyiv,
Bohdan Hawrylyshyn str. 24, Kyiv, Ukraine, 04116,
ORCID: https://orcid.org/0009-0002-3319-4502,
andrey.shvyndya22@gmail.com

A.YU. NIKOLAIENKO, Ph.D., Assistant of the Department of Software Systems and Technologies,
Taras Shevchenko National University of Kyiv,
Bohdan Hawrylyshyn str. 24, Kyiv, Ukraine, 04116,
ORCID: https://orcid.org/0000-0002-2402-2947,
n_nastja@knu.ua

PERSONALIZATION OF USER EXPERIENCE IN LANGUAGE LEARNING THROUGH LARGE LANGUAGE MODEL ASSISTANTS

Language learning benefits from a comprehensive approach, but traditional software often lacks personalization. This study analyzes prompt engineering principles to implement a test generation algorithm using Large Language Models (LLMs). The approach involved examining these principles, exploring related strategies, and creating a unified prompt structure. A test generation script was developed and integrated into an API for an interactive language learning platform. While LLM integration offers highly effective, personalized learning experiences, issues like response time and content diversity need addressing. Future advancements in LLM technology are expected to resolve these limitations.

Keywords: *Personalization, Language Learning, Artificial Intelligence, Large Language Models, Prompt Engineering, OpenAI, AI assistants.*

Introduction

Language learning is a multifaceted process that requires a holistic approach to be truly effective. Traditional software solutions often fall short in providing a comprehensive and immersive experience, as they tend to either focus solely on content delivery or adopt a one-size-fits-all educational methodology. The former approach, while potentially rich in information, lacks the interactive and adaptive elements crucial for active language acquisition. Conversely, the latter fails to account for individual learning styles, preferences, and proficiency levels, thereby limiting its efficacy.

The limitations of these conventional approaches emphasize the need for more personalized and dynamic language learning tools. Technological advancements, particularly in the field of Artificial Intelligence (AI), present promising opportunities to address this need. Large Language Models (LLMs) and AI assistants have witnessed a surge in popularity and capabilities, enabling natural language interactions and tailored content generation. However, despite their potential, there is a

Cite: Shvyndia A., Nikolaienko A. Personalization of User Experience in Language Learning through Large Language Model Assistants. *Control Systems and Computers*, 2024, 3, 68-76. https://doi.org/10.15407/csc.2024.03.068

dearth of research and practical implementations exploring the effective integration of these AI technologies into language learning systems.

Problem Statement

1. Digital Platforms for Language Learning

An imperative aspect of developing effective online language learning platforms lies in understanding the current market and identifying potential areas for improvement. Extant platforms predominantly employ standardized exercises and activities geared towards vocabulary acquisition and reinforcement [1]. However, this conventional approach may not sufficiently engage learners or cater to their individual needs and preferences. Research [2] has demonstrated the efficiency of gamification techniques in enhancing user motivation and engagement within digital learning environments. By incorporating dynamic, personalized activities tailored to individual users, platforms can foster a more immersive and engaging experience, potentially leading to improved learning outcomes. Nonetheless, manually creating such personalized content can be resource-intensive and impractical at scale. Therefore, leveraging the capabilities of LLMs and AI assistants presents a promising solution to automate content generation and adaptation based on individual user inputs and preferences.

2. Personalization as a New Studying Trend

Regarding the merits of personalized language learning approaches, it is worth examining several relevant studies. A collection of case studies by the University of Cambridge [3] demonstrates how individuals from diverse cultural backgrounds or with unique needs can achieve better outcomes through the personalization of the learning process. However, these examples relied on manual personalization, which is an inefficient and resource-intensive solution in the age of automation. A comprehensive study by the Education University of Hong Kong [4], spanning data from 2000 to 2019, revealed an emerging trend of leveraging AI technologies as a means of personalization in education. The term "AI" was employed in a broader sense within the aforementioned study than usual. The focus of this work will be specifically directed toward one aspect of AI — Large Language Models.

3. Use of Large Language Models for Learning

Among the rapidly advancing field of LLMs, ChatGPT has gained notable attention for its ability to engage in human-like dialogue and generate contextual responses based on user prompts. Various theoretical studies [5, 6] have explored the theoretical implications of leveraging such models for language learning in higher education and general contexts. They primarily focus on potential advantages, such as the generation of authentic language materials [5] or personalized learning pathways [6]. However, these studies also acknowledge inherent limitations stemming from the finite scope of pre-trained data employed in model development. While these theoretical explorations offer valuable insights for integrating LLMs, they lack sufficient empirical evidence to fully substantiate their claims.

Paper [7] provides survey results demonstrating advantages such as ease of use, perceived usefulness, and facilitated learning when employing these tools. Nonetheless, a notable gap persists in the literature — a lack of practical examples illustrating the effective integration of large language models into digital language learning platforms. Thus, it is appropriate to conduct a study aimed at bridging this gap by presenting a practical implementation of LLMs integration for personalized digital language learning experiences. This effort will advance to corroborate and extend the concepts, theories, and findings from the aforementioned studies with empirical evidence, thereby advancing the understanding and adoption of these cutting-edge technologies in the field of language education.

The Goals and Tasks of the Study

The purpose of the study is to analyze the theoretical components, methods, principles, and strategies of prompt engineering for the most effective interaction with specified Large Language Models. This will allow for the development of a test generation algorithm for a specific foreign language learning system based on the grammatical topics and words the user is studying.

To accomplish the objective, the subsequent tasks were established:

• To investigate the structure of prompts and their direct implementation using the Python programming language.

• To develop a novel algorithm for generating personalized language exercises tailored to individual learners' specific areas of study, including grammatical topics and vocabulary sets.

• To integrate the developed prompt engineering solution and personalized content generation algorithm with the existing architecture and APIs of a foreign language learning platform, enabling users to receive dynamically adapted content through LLM interaction.

Materials and Methods

The proposed test generation algorithm leverages the OpenAI API, an interface that facilitates interaction with Generative Pre-Trained Transformers (GPT models) trained to comprehend and reproduce natural language. This capability is crucial for effective prompt engineering, as it enables the development and maintenance of requests that can be easily understood and processed by the model. Important to note that the integral part of prompt engineering is the concept of a 'token' [8] — a unit of information represented by bytes with specific semantics, which serves as the fundamental object of model processing and is derived from the text provided.

Prompt engineering is a multifaceted discipline governed by several principles [8—11] that must be adhered to elicit desired responses from language models:

1. *Clarity and Concision*. Prompts should be formulated with clear and concise instructions, providing only the necessary information for the generation task. Extraneous jargon, complex language constructions, and superfluous tokens should be avoided to prevent distractions and cluttered responses. Additionally, minimizing token usage optimizes computational efficiency and cost-effectiveness when interacting with the language model.

2. *Constraints*. In certain scenarios, it may be advantageous to define explicit constraints for the model's response. By specifying limitations such as maximum word counts or content restrictions, the model's generation can be guided towards more focused and semantically appropriate outputs.

3. *Task Decomposition*. Complex generation tasks often encompass multiple subtasks. In the context of test creation, for instance, these subtasks may include defining the test task, generating answer options, and identifying the correct responses. Decomposing these tasks into smaller, more manageable components can prevent token mixing and loss of context, thereby enhancing the model's ability to maintain semantic coherence and produce more accurate results.

By adhering to these principles, prompt engineering can be leveraged to optimize the interaction between language models and the proposed test generation algorithm, enabling the creation of personalized and effective language learning assessments.

In addition to the core principles of prompt engineering, it is imperative to employ strategic methods that further refine the structure and content of prompts to elicit optimal responses from language models [8, 9]. One such approach is *role-playing*, where the model is assigned a specific persona or perspective to adopt. By explicitly defining the context and constraints within which the model should operate, the semantic search space is effectively narrowed, preventing the consideration of irrelevant domains that could yield unexpected or undesirable outputs. Instead, the model's generation is confined to a controlled and focused domain, enhancing the relevance and coherence of the results.

Another valuable strategy is *few-shot prompting*, also known as in-context learning. This technique involves providing the model with a set of exemplar inputs and outputs that represent the desired response format and structure (e.g., JSON). By exposing the model to these examples, its generated responses are more likely to conform to the specified format, streamlining subsequent processing and integration with the target application. Moreover, the inclusion of well-crafted example responses further constrains the semantic scope of the model, guiding it toward generating content that closely aligns with the provided exemplars and the intended use case.

Collectively, these strategies complement the core principles of prompt engineering, enabling a more controlled and directed interaction with language models. In alignment with established principles, the proposed prompt structure comprises the following elements:

1. *Persona Definitio*n. To delineate the semantic domain and establish context, the language model is assigned the persona of a "language expert" with the specific purpose of "providing personalized language assessments". By employing the role-playing strategy, the model is primed to generate linguistically oriented test items, as its knowledge base is confined to the domain of a specialized language professional.

2. *Context Specification*. The prompt incorporates contextual parameters that define the linguistic environment in which the test generation occurs. This includes identifying the target language, as well as any specific grammatical topics or vocabulary sets that should be covered in the generated assessments. By providing this contextual information, the model can tailor its outputs to align with the specified linguistic constraints.

3. *Task Decomposition*. The overarching task of test generation is broken down into a series of subtasks. These subtasks encompass the selection of grammatical themes and vocabulary items, the formulation of sentences, the construction of questions based on those sentences, the articulation of task prompts, and the generation of answer choices. Delineating these subtasks ensures a structured and organized approach to the test generation process.

4. *Output Format Definition*. To ensure compatibility with the target language learning platform, the prompt specifies the expected output format as a JSON object, outlining the required attributes and their corresponding data types. This format aligns with the existing data structure used by the platform's test model. Additionally, certain attributes are explicitly constrained based on the execution context, such as the unique identifier, the generation status, the task type, the answer type, and the resource identifier.

5. *Example Provision*. Leveraging the few-shot prompting strategy, the prompt includes a representative example of a test item sourced from the platform's database. This exemplar serves as a guiding reference, directing the model's generation towards outputs that closely resemble the desired format and structure, thereby further constraining the semantic search space.

By integrating these components into a cohesive prompt structure, the proposed approach aims to optimize the interaction between the language model and the test generation algorithm. This structured approach facilitates the creation of personalized and effective language assessments that align with the specified linguistic contexts, adhere to the desired output formats, and exhibit the characteristics of high-quality test items.

Results of the Study

1. Implementation of the Generation Script

Adhering to the established principles and strategies for prompt engineering, the generation script incorporates a meticulously crafted prompt that encompasses the requisite structural components (Table 1). This prompt, whose composition aligns with the theoretical framework outlined previously, serves as a blueprint for eliciting the desired responses from the language model.

The prompt functions as the primary input to the OpenAI API, initiating the content generation process upon each API call. The resulting output, obtained through the API's response, undergoes subsequent parsing and processing to be seamlessly integrated into the algorithm's execution flow, ultimately yielding the final output.

2. Integration of the Script to the API

By integrating this Python script to the system API, the unified method for test generation is available, so that the users can get the personalized activities based on their contextual input. For example, there are such scenarios:

- "Text-Select" type with both grammar topics and words specified (Fig. 1) — answers incorporate both grammar topics and vocabulary comprehension.

• “Audio-Input” type with grammar topic only (Fig. 2) — the resource file is generated and the task incorporates grammar comprehension.

• “Text-Audio” type with vocabulary topic only (Fig. 3) — no answers are generated and the task incorporates vocabulary comprehension.

Table 1. **Generation prompt structure**

Structural component	Value
Persona	You are a language expert for a language learning app, whose main goal is to provide a personalized test for a specific user based on the grammar topics and words the user wants to learn in a specific language.
Context	Now, there is a user, who learns the {language} language. “Their grammar topics to learn: {‘, ‘.join (grammar_topics)}” if (number_of_grammar_topics > 0) else “The are no grammar topics to learn”. “Their words to learn: {‘, ‘.join(words)}” if (number_of_words > 0) else “There are no words to learn”.
Task	Your task is to create a test for the user according to such steps: 1. Select a grammar topic from the list above (if there are any). 2. Select a word from the list above (if there are any). 3. Invent a generic SENTENCE using the selected grammar topic and a word. 4. Specify some linguistic QUESTION in the specified language to this sentence. 5. Now, COMBINE the sentence and the question. THE TASK MUST INCLUDE BOTH THE SENTENCE AND THE QUESTION. Check the example again before generating a response. 6. Specify the correct answer to this question{‘ and specify 3 more incorrect answers’ if is_answer_select else ‘’}. The answer should be a word or phrase. The answer should be in the language the user is learning.
Output format	Your output should be a JSON in such structure: id: Number (ALWAYS 0) language: String (ALWAYS {language_code}) isGenerated: Boolean (ALWAYS true) category: nested json object with fields “grammarTopics” and “words” taskType: String (ALWAYS {task_type}) answerType: String (ALWAYS {answer_type}) task: String (“‘Sentence’ Question?”) answers: [String] {‘’ if is_answer_select else ‘empty list for this case’} correctAnswer: String resource: String (ALWAYS null) NOTHING ELSE SHOULD BE IN YOUR OUTPUT

Example	For example,
	{
	“id”: 0,
	“isGenerated”: true,
	“language”: “en”
	“category”: {
	“grammarTopics”: [“past simple”, “present continuous”],
	“words”: [“apple”, “republic”]
	},
	“taskType”: “text”,
	“answerType”: “select”,
	“task”: “‘I ate an apple yesterday.’ Which tense is it?”,
	“answers”: [“past simple”, “present continuous”, “future perfect”, “present simple”],
	“correctAnswer”: “past simple”,
	“resource”: null
	}

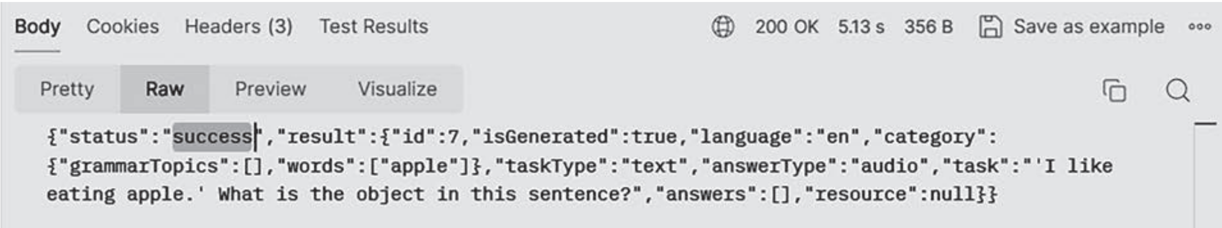

Fig. 1. Test generation for the “Text-Select” type with “Past simple” and “Apple” as categories

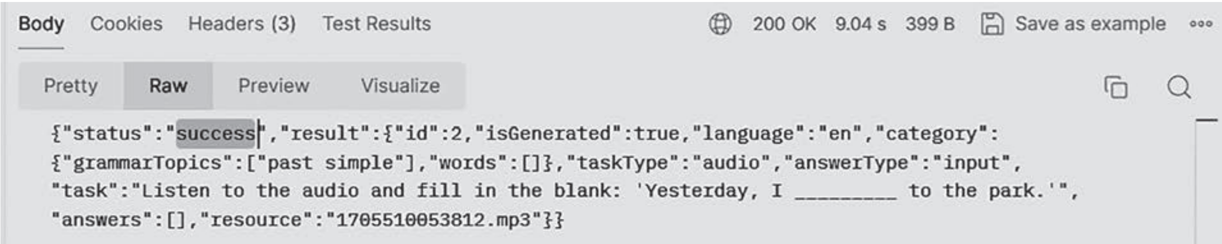

Fig. 2. Test generation for the “Audio-Input” type with only “Past Simple” as a category

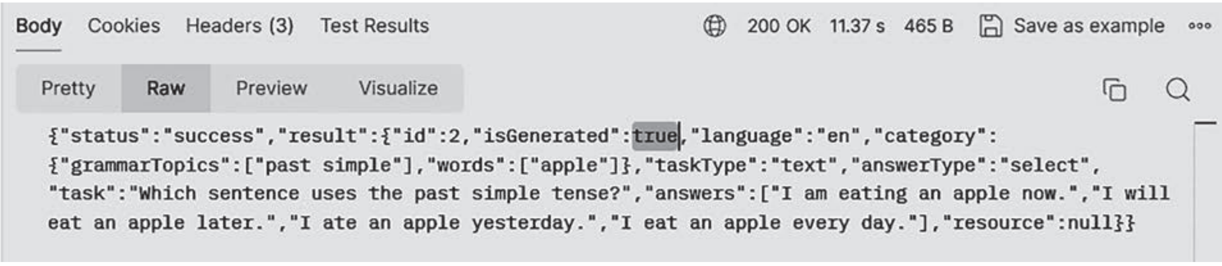

Fig. 3. Test generation for the “Text-Audio” type with only “Apple” as a category

In other words, no matter the contextual input provided, the unified model, which corresponds to the grammar topics and words provided, is returned.

3. Integration into the Existing System

The proposed approach was successfully implemented through a client-server architecture, as depicted in Fig. 4. This architecture facilitated a seamless data flow, encompassing the API calls to the language model, the processing of the generated responses, and their subsequent rendering on the user interface of the language learning platform.

Consequently, an interactive and immersive platform for foreign language acquisition was realized. This platform features a comprehensive and efficient interface that enables users to generate personalized learning activities tailored to their individual needs and proficiency levels. By harnessing the capabilities of large language models, the platform democratizes access to adaptive and engaging language learning experiences, empowering a diverse range of learners to embark on their linguistic journeys.

Results Discussion

The results demonstrate the potential for seamlessly integrating LLMs into digital language learning platforms to provide personalized user experiences tailored to individual contextual inputs. Notably, the OpenAI API has proven to be an accessible and straightforward interface for integrating LLM capabilities into existing systems. The implementation process is streamlined, requiring

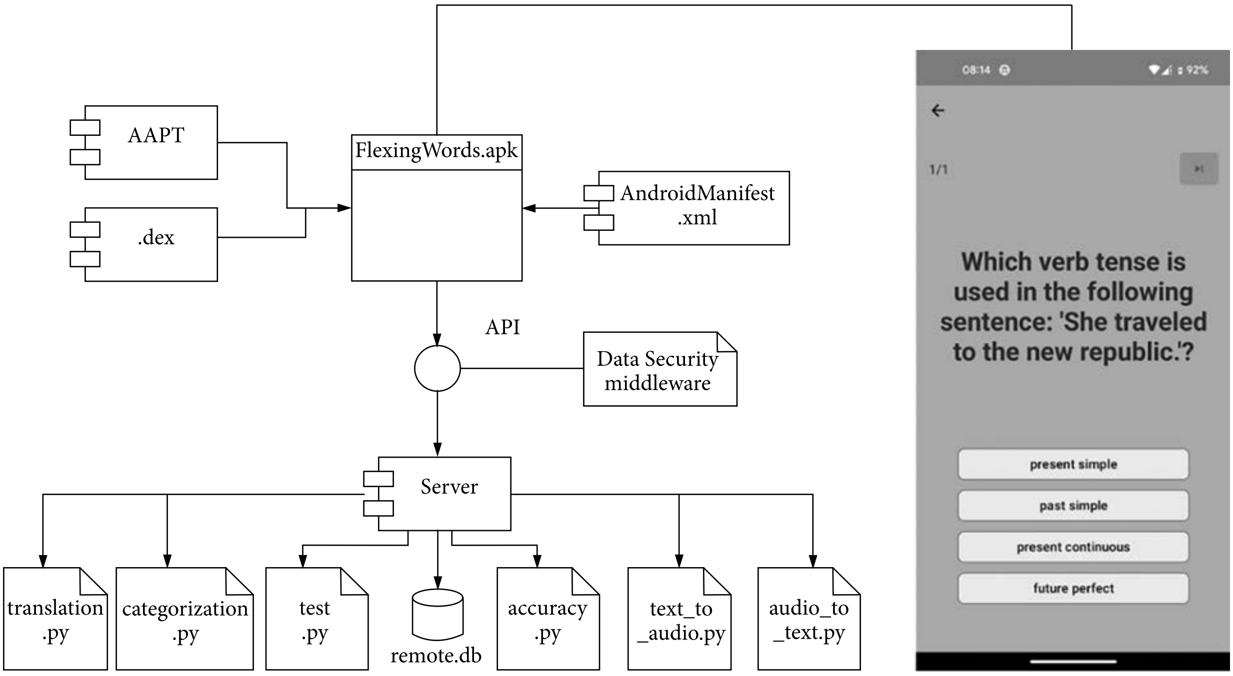

Fig. 4. System Architecture

only the creation of a string-based prompt and a single method call to initiate the content generation process.

The generated content obtained through API calls exhibits a high degree of relevance and coherence, closely aligned with the specified input parameters. For instance, when both a grammatical topic and vocabulary set are provided as inputs, the resulting content effectively incorporates elements from both domains. Conversely, when only one input type is specified, the generated content focuses exclusively on that linguistic aspect. This flexible and context-driven approach empowers users to receive personalized learning activities and assessments tailored to their specific areas of study, fostering a more engaging and effective educational experience.

However, it is imperative to acknowledge and address certain limitations inherent to the current implementation. One notable concern is the response time, which ranges from 5 to 11 seconds, potentially hindering real-time interactivity in certain system contexts. Additionally, despite the extensive training data utilized by the LLM, its knowledge base remains finite. Consequently, there is a potential limitation on the diversity and novelty of the generated content, especially over prolonged periods of extensive use. Nonetheless, these limitations are intrinsically tied to the current state of LLM technology and are not inherent flaws of the proposed integration approach.

It is worth noting that the observed constraints on content diversity may be mitigated by providing a sufficiently diverse range of input categories, thereby expanding the potential for unique and varied outputs. Furthermore, as LLM technology continues to advance, future iterations of models like GPT are expected to address these performance and scalability concerns more effectively.

While the current study provides a solid foundation for LLM integration in language learning platforms, additional research is warranted to further optimize the proposed approach and address remaining limitations. Potential areas of exploration include strategies for improving response times, techniques for augmenting training data to enhance content diversity, and methodologies for ensuring the long-term scalability and sustainability of LLM-powered personalized learning experiences.

Conclusion

The findings of this study underscore the immense potential of integrating open-source large language models (LLMs) into digital language learning platforms. By leveraging the dynamic content generation capabilities of these models, educational platforms can cultivate highly interactive and personalized learning experiences meticulously tailored to individual needs and contexts. Moreover, the automated nature of LLM-driven content generation alleviates the resource-intensive burden of manual content curation. Consequently, language learning platforms remain perpetually enriched with novel and diverse educational materials.

The implementation described in this study highlights the relative ease of integrating LLM capabilities into existing platforms, thereby minimizing barriers to adoption. Furthermore, the cost-effectiveness of leveraging open-source language models presents a significant advantage, potentially broadening access to advanced language learning technologies across a wide range of educational settings.

REFERENCES

1. Bączkowska, A. (2021). "An Overview of Popular Website Platforms and Mobile Apps for Language Learning." Forum Filologiczne Ateneum, 1(9), pp. 9—35, https://doi.org/10.36575/2353-2912/1(9)2021.009.
2. Arce, N.H., & Valdivia, A.C. (2020). "Adapting competitiveness and gamification to a digital platform for foreign language learning". *International Journal of Emerging Technologies in Learning* (*iJET*), 15.20, pp. 194—209.
3. Kukulska-Hulme, A. (2016). *Personalization of language learning through mobile technologies*. Cambridge University Press, Cambridge, UK.
4. Chen, X., Zou, D., Xie, H., & Cheng, G. (2021). "Twenty years of personalized language learning". *Educational Technology & Society*, 24(1), pp. 205—222.
5. Baskara, R. (2023). "Exploring the implications of ChatGPT for language learning in higher education". *Indonesian Journal of English Language Teaching and Applied Linguistics, 7*(2), pp. 343—358.
6. Hatmanto, E. D., & Sari, M. I. (2023). "Aligning Theory and Practice: Leveraging Chat GPT for Effective English Language Teaching and Learning". *In E3S Web of Conferences*, Vol. 440, p. 05001.
7. Shaikh, S., Yayilgan, S.Y., Klimova, B., Pikhart, M. (2023). "Assessing the Usability of ChatGPT for Formal English Language Learning". *Eur. J. Investig. Health Psychol. Educ*., 13, pp. 1937—1960.
8. Detailed guide to Prompt Engineering from Matt Crabtree. [online]. Available at: <https://www.datacamp.com/blog/what-is-prompt-engineering-the-future-of-ai-communication> [Accessed 11 May 2024].
9. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., ... & Schmidt, D.C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382. Cornell University, February 2023.
10. Ekin, S. (2023). "Prompt engineering for ChatGPT: a quick guide to techniques, tips, and best practices". Authorea Preprints. May 04.
11. Bozkurt, A., & Sharma, R. C. (2023). "Generative AI and Prompt Engineering: The Art of Whispering to Let the Genie Out of the Algorithmic World". *Asian Journal of Distance Education*, 18(2), pp. I—VII.

Received 04.06.2024

А.М. Швиндя, студент,
Київський національний університет України імені Тараса Шевченка,
вул. Богдана Гаврилишина, 24, м. Київ, Україна, 04116,
ORCID: https://orcid.org/0009-0002-3319-4502,
andrey.shvyndya22@gmail.com
А.Ю. Ніколаєнко, к.т.н., асистент,
Київський національний університет України імені Тараса Шевченка,
вул. Богдана Гаврилишина, 24, м. Київ, Україна, 04116,
ORCID: https://orcid.org/0000-0002-2402-2947,
n_nastja@knu.ua

ПЕРСОНАЛІЗАЦІЯ ДОСВІДУ КОРИСТУВАЧА У ВИВЧЕННІ ІНОЗЕМНИХ МОВ ЗАВДЯКИ ВЕЛИКИМ МОВНИМ МОДЕЛЯМ

Вступ. Вивчення мов потребує комплексного підходу для досягнення ефективності. Традиційні програмні рішення часто не виправдовують очікувань, оскільки або занадто зосереджуються на наданні контенту, або використовують універсальний підхід. Це може обмежувати учня через неврахування його індивідуального стилю навчання. Зростає потреба в персоналізованих інструментах для вивчення мов. Штучний інтелект, особливо великі мовні моделі, має великий потенціал для досягнення необхідного рівня персоналізації.

Мета: аналіз методів, принципів і стратегій інженерії запитів (промптів) для ефективної взаємодії з великими мовними моделями. Це дає змогу розробити алгоритм генерації тестів на основі контекстуального введення користувача.

Методи. Використано систематичний підхід до розроблення алгоритму генерування тестів, починаючи з аналізу ключових принципів інженерії запитів із подальшим розглядом стратегій на основі цих принципів. Останнім етапом є створення узгодженої структури промпту на основі визначених принципів і стратегій.

Результати. Розроблено та інтегровано до *API* скрипт для генерування тестів, що став складовою інтерактивної платформи для вивчення іноземних мов.

Висновки. Інтеграція *LLM* у платформи для вивчення мов може зі значною ефективністю забезпечити персоналізовані та залежні від контексту користувача навчальні процеси, хоча й поточні обмеження включають затримки у часі відповіді та частково обмежену різноманітність контенту. Очікується, що майбутні вдосконалення технологій асистентів *LLM* сприятимуть вирішенню цих проблем.

Ключові слова: *персоналізація, вивчення іноземних мов, штучний інтелект, великі мовні моделі, інженерія промптів, OpenAI, асистенти ШІ.*