Application of neural networks for adaptive and flexible electronic tourist guide

Artur Moroz[1], Illia Solohubov[1], Mariia Yu. Tiahunova[1], Halyna H. Kyrychek[1] and Stepan Skrupsky[1]

[1]National University "Zaporizhzhya Polytechnic", 64 Zhukovskyi Str., Zaporizhzhya, 69063, Ukraine

Abstract

In this paper the utilisation of neural networks is examined, focusing on an electronic tourist information query system as an example. The analysis begins with a review of the fundamental principles underlying the operation of neural networks, their distinctions from traditional programming, and the reasons they can be beneficial in such systems. Additionally, the work of OpenAI and their development of ChatGPT are considered, illustrating the potential applications of neural networks in real-world scenarios. The paper compares the capabilities of neural networks with dynamic parameter sets against hardcoded logic. An analysis of the impact of these approaches on the speed of system development and the number of possible parameter combinations is conducted. The study also describes the implementation of neural networks on the Node.js platform, demonstrating the practical aspects of this approach, and analysing specific examples of responses and reactions to unforeseen queries for which programming a response is unfeasible. The study includes an analysis of existing query systems and the development and implementation of a prototype system. Moreover, the paper encompasses a comparative analysis of experiment results. The findings corroborate the advantages of neural networks over traditional hardcoded logic, contributing to the enhancement of tourist services in the digital age. .

Keywords

neural networks, OpenAI, ChatGPT, survey system, electronic guide tourist

1. Introduction

In the contemporary digital world, adaptive artificial intelligence technologies are becoming an increasingly integral part of our lives. The development of efficient and flexible artificial intelligence systems capable of executing complex tasks is imperative for the progression of various facets of our existence. One such domain is the electronic tourist information query systems, where the employment of neural networks proves to be an efficacious alternative to conventional programming [1].

The tourism sector of the modern world is increasingly harnessing digital technologies with the aim of enhancing customer service quality and ensuring maximum individualisation of the user experience. Tourist information query systems serve as a crucial tool for gathering data pertaining to the needs and desires of visitors, as well as providing information that meets these needs [2].

Traditionally, such systems have been developed employing hardcoded logic. However, they can be rigid and non-adaptive as they are constrained by pre-defined response options and interactions that were anticipated and programmed a priori. This leads to the system's inadequacy in responding to unforeseen replies or scenarios that were not accounted for in the programming logic.

The application of neural networks can be a potential solution to this issue. Neural networks are capable of learning from data and adapting to new situations, as opposed to adhering to a rigid set of rules [3]. This enables the electronic tourist information query systems to be more pliable and better equipped to fulfil user needs.

However, the utilisation of neural networks is accompanied by its own set of challenges [4]. Among these are concerns regarding the speed of system development, the quantity of possible parameter com-

CTE 2023: 11th Workshop on Cloud Technologies in Education, December 22, 2023, Kryvyi Rih, Ukraine
arthur.official.moroz@gmail.com (A. Moroz); illia.solohubov@gmail.com (I. Solohubov); mary.tyagunova@gmail.com (M. Yu. Tiahunova); kirgal08@gmail.com (H. H. Kyrychek); sskrupsky@gmail.com (S. Skrupsky)
0009-0008-6742-2298 (A. Moroz); 00009-0000-6140-3485 (I. Solohubov); 0000-0002-9166-5897 (M. Yu. Tiahunova); 0000-0002-0405-7122 (H. H. Kyrychek); 0000-0002- 9437-90 (S. Skrupsky)

binations, and the handling of unanticipated responses [5, 6]. Hence, it is apposite to conduct research aimed at ascertaining the potentials of employing neural networks to heighten the efficacy of such systems and to delineate the advantages of this approach compared to traditional logic programming.

The investigation into the application of neural networks in tourist information query systems and related areas provides critical context and foundational knowledge for the advancement of this field, aiding in understanding the current state of research and in identifying prospects for further investigation.

Aljanabi [7] considers the prospects and capabilities of employing artificial intelligence models, specifically ChatGPT, for enhancing interaction with users and the development of electronic tourist information query systems. This work unveils new horizons in comprehending the potential of artificial intelligence models in the tourism industry.

Bughin [8] examines the application of artificial intelligence models, particularly ChatGPT, in various domains, including the tourism sector. The author conducts an analysis of the advantages and limitations of such applications, which allows for the identification of optimal scenarios for the use of artificial intelligence models in tourist information query systems.

Wang et al. [9] explore an approach in which neural networks are combined with A* search algorithms for personalised route recommendations. This study reveals the potential of neural networks in improving search algorithms and presents new possibilities for creating flexible and adaptive tourist information query systems.

Wang et al. [10] propose an improved route search algorithm using neural networks for personalized recommendations. This paper examines the use of neural networks in conjunction with route search algorithms, contributing to the enhancement of efficiency and accuracy in tourist information query systems.

Mou et al. [11] proposes a model for personalized tourist route recommendations employing neural networks and trajectory understanding. This work investigates ways of using neural networks to improve the accuracy and personalisation of tourist route recommendations.

The application of neural networks has significant advantages over traditional logic programming, especially when compared with systems with a dynamic set of parameters [12, 13, 14].

It's important to note that, although neural networks is capable of generating persuasive and informative responses, it still relies on statistical correlations in the data it has been trained on and does not possess consciousness or understanding in the traditional sense of these terms.

2. Application of neural network in TypeScript

TypeScript play a pivotal role in the modern software development landscape, facilitating the transformation of both web technologies and general-purpose development [15, 16].

Developed by OpenAI, GPT-3.5-turbo is a cutting-edge machine learning model that is built upon the principles of the GPT-3 architecture and specialises in text generation, simulating human language. A standout feature of this model is its over 175 billion parameters, which endow GPT-3.5-turbo with the capability to effectively model responses that adapt to specific contexts based on input data analysis.

Integrating GPT-3.5-turbo with Node.js and TypeScript can assist in developing powerful, adaptable, and efficient applications. This can encompass text generation, automated content moderation, recommendation system development, chatbot creation, and more [17, 18].

To commence the integration of GPT-3.5-turbo into a JavaScript/TypeScript application, the OpenAI API must be integrated. This may necessitate the installation of the 'openai' package, which serves as OpenAI's official client library for NPM, and simplifies interaction with GPT-3.5-turbo within the application.

It is worth mentioning that utilizing this package is recommended, yet not obligatory. To retrieve information from the OpenAI API, one has the option to manually compose a request using the 'axios' package or even the native methodologies inherent to the JavaScript programming language. This entails specifying the appropriate endpoint, setting the necessary headers, and formatting the data correctly. The 'openai' package essentially streamlines and optimizes this procedure.

To utilize alternative Large Language Models [19], one must only modify the query parameters. In the current configuration, we employ the 'GPT-3.5-turbo model'. However, to transition to another model, take the hypothetical 'gpt-4' as an example, one would simply adjust the model's identifier in the text constant prior to initiating the request. It is noteworthy that, in this illustrative example, the input parameter schema for the speculative 'gpt-4' remains congruent with that of GPT-3.5-turbo.

While there are other accessible language models like 'davinci', the input data format remains largely consistent. Nevertheless, due to their comparative limitations, it is prudent to exercise caution when assessing the capabilities of these models in relation to the system under discussion.

With TypeScript, it is possible to craft clear and secure interfaces and types reflecting the structure of GPT-3.5-turbo's API queries and responses. TypeScript affords convenient tools for static typing, which can enhance the quality of your code whilst also facilitating comprehension and debugging.

Figure 1 illustrates the configuration and utilisation of the 'openai' package with the TypeScript programming language for creating a class instance. Figure 2 presents an example of using configuration from figure 1 to perform a query.

```
import { Configuration, OpenAIApi } from 'openai';
import { ChatCompletionRequestMessage } from 'openai/api';

const configuration: Configuration = new Configuration( param: {
    organization: 'ORG-Key',
    apiKey: 'API-Key'
});
2 usages
const openai: OpenAIApi = new OpenAIApi( configuration: configuration);

1 usage
const templateMessage: string = 'Your Template Message';

2 usages
const message: ChatCompletionRequestMessage = {
    role: 'user',
    content: templateMessage
};

export { openai, message, templateMessage };
```

Figure 1: Configuration and use of the 'openai' package with the TypeScript programming language.

3. Analysis of using a neural network for generating responses

The neural network has a large number of possible interpretations of the responses obtained from training, allowing it to utilise natural language when communicating with the user. However, to use this in place of programmed logic, the neural network of such a model requires a well-crafted and accurate formulation. This formulation will be referred to as the base or template message.

For employing the neural network for the purpose of searching for countries based on an arbitrary number of dynamic parameters, the template message will look as shown in figure 3.

The parameters themselves will be added line by line to the template message.

After two parameters were added to the template message, the final version of the query will read as follows:

Countries need to be found that:

```
try {
    const response = await openai.createChatCompletion(
        {
            model: 'gpt-3.5-turbo',
            messages: [message],
            temperature: 0.2
        }
    );
    const array = JSON.parse( response.data.choices[0].message.content);
    const countries = array.map((item) => CountriesProvider.getCountryByCode2( item)).filter((item) => !!item);
    console.log(countries);
} catch (e) {
    console.log(e);
}
```

Figure 2: Example of using configuration from figure 1 to perform a query.

```
Chat, I need you to perform the following tasks without any interpretation.
I will give you a list of task parameters and you will return single  array of ISO countries with
no other text or desctiptionalways excluding RU, RB. Give answer only as array . If answer is
null - return null or empty array. If you don't have data for any of the parameters - use
archived data.
Task: what are the countries
nearest to latitude: 47.901      ,longitude: 35.143      .
and temperature is higher 50 Celsius

["KZ", "AZ", "IR", "SA", "IQ", "KW", "AE"]
```

Figure 3: Template message for searching for countries based on an arbitrary number of dynamic parameters.

1. Are closest to a certain GPS location.
2. Have a temperature above 50 degrees Celsius.

The result of executing the query with these parameters is shown in figure 4.

```
Chat, I need you to perform the following tasks without any interpretation.
I will give you a list of task parameters and you will return single  array of ISO countries with
no other text or desctiptionalways excluding RU, RB. Give answer only as array . If answer is
null - return null or empty array. If you don't have data for any of the parameters - use
archived data.
Task: what are the countries
nearest to latitude: 47.901      ,longitude: 35.143      .
and temperature is higher 50 Celsius

["KZ", "AZ", "IR", "SA", "IQ", "KW", "AE"]
```

Figure 4: Use of template message with two parameters.

In response, an array of ISO (Alpha2) country codes was received, which meet the specified conditions. The next set of parameters is similar to the previous set1, only instead of finding countries with a temperature above 50 degrees Celsius, the system received a parameter for searching for countries with a temperature range from 20 to 50 degrees Celsius. The result of executing the query with these parameters is shown in figure 5.

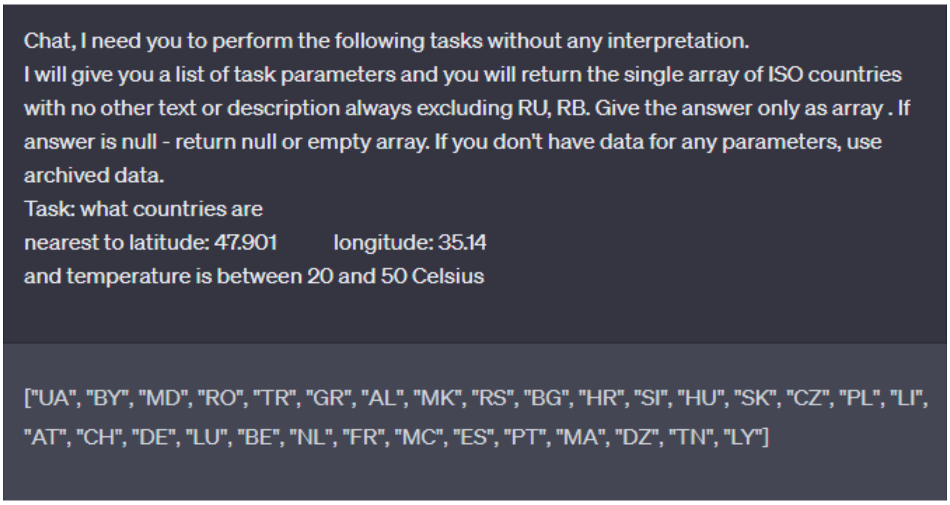

Figure 5: Use of template message with two parameters.

In response, an array of ISO (Alpha2) country codes was also received, which meet the specified conditions.

Next query includes an additional parameter for refinement, namely a parameter where the currency in the country is the Euro. The result of executing the query with these parameters is shown in figure 6.

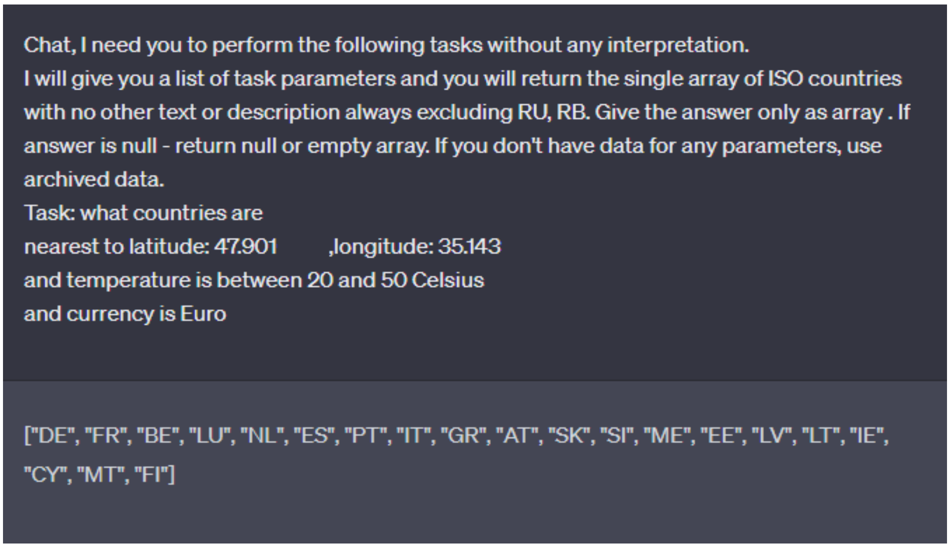

Figure 6: Use of template message with three parameters.

In response, an array of ISO (Alpha2) country codes was also received, which meet the specified conditions.

In the next query, only one temperature parameter was used, which reduces the number of search parameters and increases the number of potentially received parameters. The result of executing the query with this parameter is shown in figure 7.

As can be seen from the response, the array of ISO (Alpha2) country codes, which meet the specified conditions, is larger because there were fewer selection criteria.

In the mext query, an additional parameter was added, specifying that the climate in the country should be tropical. Using programmed logic to answer this query, a new field called “climate” would

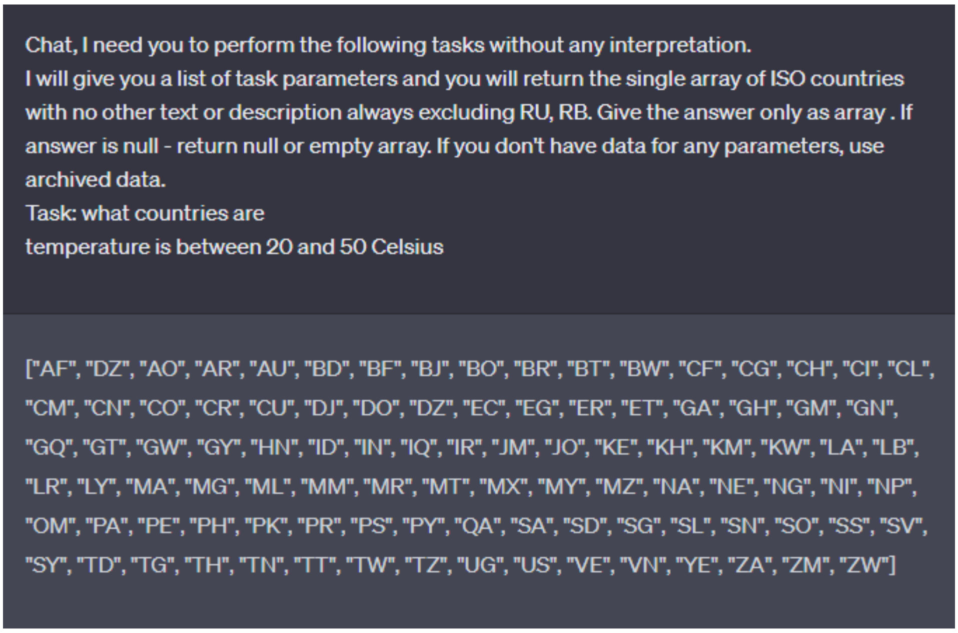

Figure 7: Use of template message with one parameter.

need to be added to the database and filled for each country (193±2). The neural network, on the other hand, does not require any additional configurations to respond to this query. The result of executing the query with this parameter is shown in figure 8.

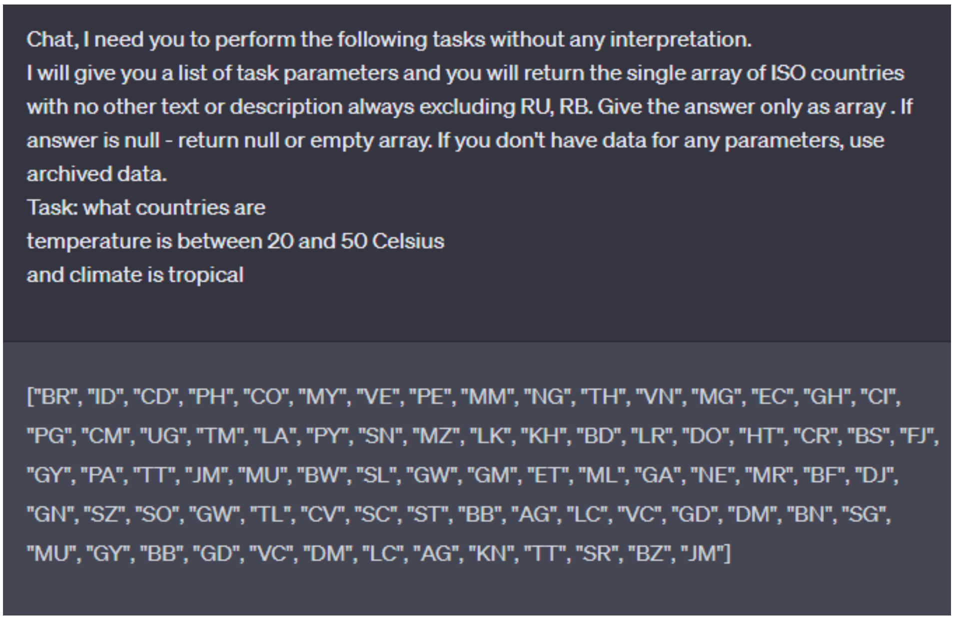

Figure 8: Use of template message with two parameters.

As can be seen from the response, the array of ISO (Alpha2) country codes that meet the specified conditions is smaller compared to example 4 because an additional query parameter has appeared. It is also noticeable that some countries, such as Afghanistan or Tanzania, which are not tropical, are absent, while some countries, such as India or Argentina, are present in both responses.

Unlike the previous examples, a parameter has been added, specifically the presence and absence of terrorist attacks during a certain period, predicting which at the design stage is not possible. The result of executing queries with these parameters is shown in figure 9 and figure 10.

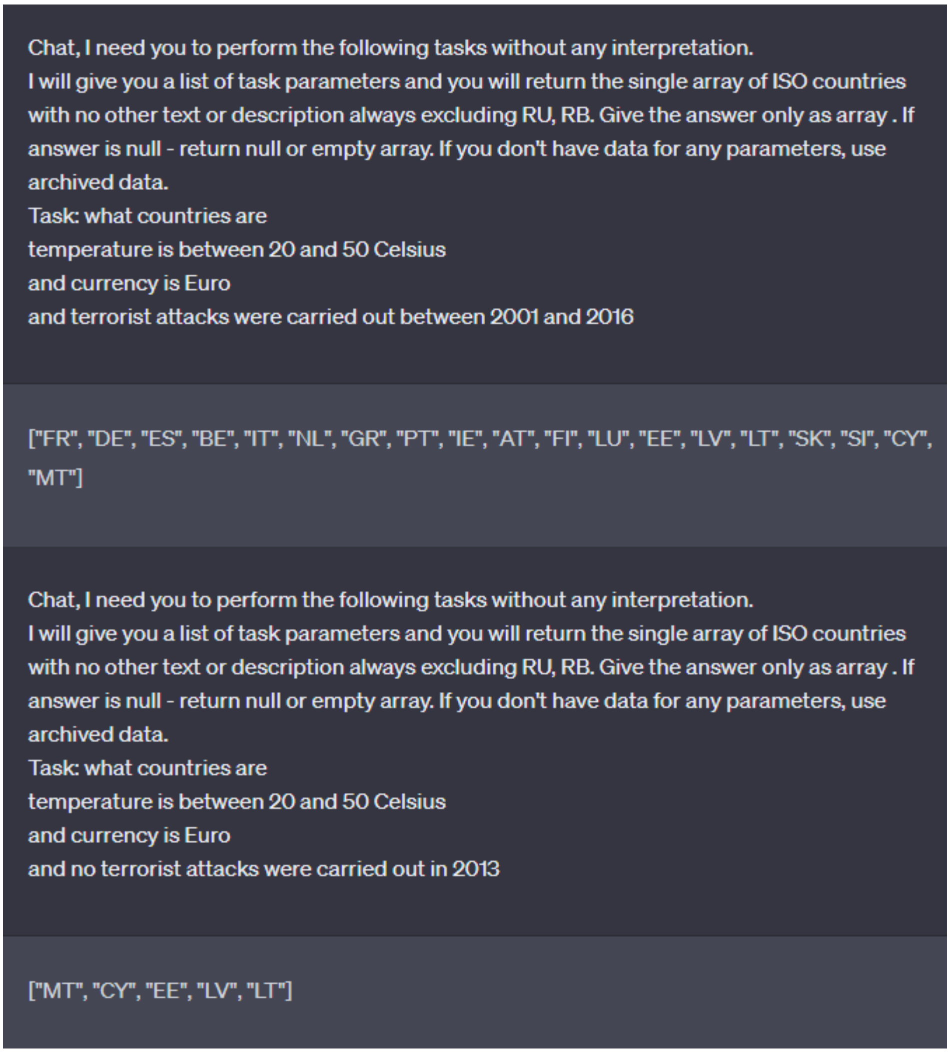

Figure 9: Use of template message with unpredictable parameters.

4. Conclusions

Despite the evident advantages highlighted in the previous examples, particularly the ability to execute queries with dynamic parameters, which is unattainable for systems with programmed logic, it is necessary to consider that the development of a specific neural network can be a significant expenditure and labour-intensive process. Furthermore, employing open neural networks, such as GPT-3.5-turbo, may be most prudent in situations where user queries do not require the processing of confidential or corporate information. Therefore, the decision to utilise neural networks should be well-founded, taking into account the specific requirements and constraints of the project, including factors such as data protection, budget, timeframes, and technical resources.

Examining metrics such as accuracy and total query execution time may only provide a cursory understanding, given that the two juxtaposed approaches present distinct advantages and disadvantages. Neural networks, for instance, might exhibit a reduced nominal accuracy, and their command execution time is predominantly governed by request execution durations (typically ranging from 3-5 seconds). However, they afford an expedited implementation speed, approximately 2-3 orders of magnitude swifter, and remain uninhibited by the constraints typical of standard server APIs.

It is important to note that within the context of a survey system of an electronic directory for tourists, which typically operates with open and publicly available data that do not require specific measures regarding confidentiality, the use of a neural network such as GPT-3.5-turbo, can prove to

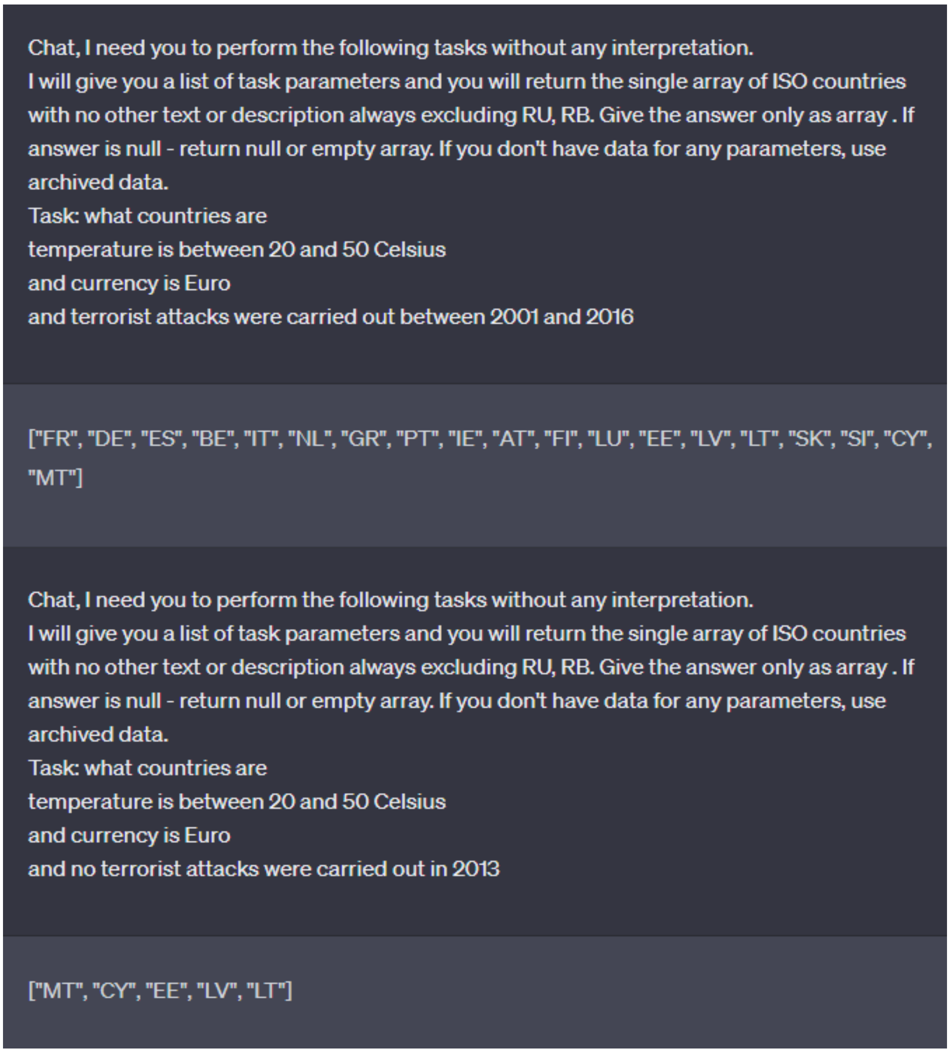

Figure 10: Use of template message with unpredictable parameters.

be extraordinarily productive. This allows for the efficient processing of user queries and promptly providing responses. Such an approach can significantly enhance the efficiency of the system, providing users with more relevant and substantial information within concise time intervals.

References

[1] M. P. Shyshkina, M. V. Marienko, The use of the cloud services to support the math teachers training, CTE Workshop Proceedings 7 (2020) 690–704. doi:10.55056/cte.419.

[2] E. A. Kosova, Distance course "Information systems and technology" for speciality "Tourism", CTE Workshop Proceedings 2 (2014) 213–224. doi:10.55056/cte.211.

[3] S. O. Semerikov, T. A. Vakaliuk, I. S. Mintii, V. A. Hamaniuk, V. N. Soloviev, O. V. Bondarenko, P. P. Nechypurenko, S. V. Shokaliuk, N. V. Moiseienko, V. R. Ruban, Development of the computer vision system based on machine learning for educational purposes, Educational Dimension 5 (2021) 8–60. doi:10.31812/educdim.4717.

[4] I. A. Pilkevych, D. L. Fedorchuk, M. P. Romanchuk, O. M. Naumchak, Approach to the fake news detection using the graph neural networks, Journal of Edge Computing 2 (2023) 24–36. doi:10.55056/jec.592.

[5] A. V. Ryabko, O. V. Zaika, R. P. Kukharchuk, T. A. Vakaliuk, V. V. Osadchyi, Methods for predicting

the assessment of the quality of educational programs and educational activities using a neuro-fuzzy approach, CTE Workshop Proceedings 9 (2022) 154–169. doi:10.55056/cte.112.

[6] O. Pronina, O. Piatykop, The recognition of speech defects using convolutional neural network, CTE Workshop Proceedings 10 (2023) 153–166. doi:10.55056/cte.554.

[7] M. Aljanabi, ChatGPT: Future Directions and Open possibilities, Mesopotamian Journal of CyberSecurity 2023 (2023) 16–17. doi:10.58496/MJCS/2023/003.

[8] J. Bughin, To ChatGPT or not to ChatGPT: A note to marketing executives, Applied Marketing Analytics 9 (2023) 110–116. doi:10.2139/ssrn.4411051.

[9] J. Wang, N. Wu, W. X. Zhao, F. Peng, X. Lin, Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 539–547. doi:10.1145/3292500.3330824.

[10] J. Wang, N. Wu, W. X. Zhao, Personalized Route Recommendation With Neural Network Enhanced Search Algorithm, IEEE Transactions on Knowledge and Data Engineering 34 (2022) 5910–5924. doi:10.1109/TKDE.2021.3068479.

[11] N. Mou, Q. Jiang, L. Zhang, J. Niu, Y. Zheng, Y. Wang, T. Yang, Personalized tourist route recommendation model with a trajectory understanding via neural networks, International Journal of Digital Earth 15 (2022) 1738–1759. doi:10.1080/17538947.2022.2130456.

[12] M. Tiahunova, O. Tronkina, G. Kirichek, S. Skrupsky, The Neural Network for Emotions Recognition under Special Conditions, in: S. Subbotin (Ed.), Proceedings of The Fourth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2021), Zaporizhzhia, Ukraine, April 27, 2021, volume 2864 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 121–134. URL: https://ceur-ws.org/Vol-2864/paper11.pdf.

[13] H. Kravtsov, A. Pulinets, Interactive Augmented Reality Technologies for Model Visualization in the School Textbook, in: O. Sokolov, G. Zholtkevych, V. Yakovyna, Y. Tarasich, V. Kharchenko, V. Kobets, O. Burov, S. Semerikov, H. Kravtsov (Eds.), Proceedings of the 16th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer. Volume II: Workshops, Kharkiv, Ukraine, October 06-10, 2020, volume 2732 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 918–933. URL: https://ceur-ws.org/Vol-2732/20200918.pdf.

[14] S. Papadakis, A. E. Kiv, H. M. Kravtsov, V. V. Osadchyi, M. V. Marienko, O. P. Pinchuk, M. P. Shyshkina, O. M. Sokolyuk, I. S. Mintii, T. A. Vakaliuk, L. E. Azarova, L. S. Kolgatina, S. M. Amelina, N. P. Volkova, V. Y. Velychko, A. M. Striuk, S. O. Semerikov, ACNS Conference on Cloud and Immersive Technologies in Education: Report, CTE Workshop Proceedings 10 (2023) 1–44. doi:10.55056/cte.544.

[15] N. A. Kharadzjan, Formation of cloud-based learning environment for professional training in programming, CTE Workshop Proceedings 2 (2014) 263–268. doi:10.55056/cte.216.

[16] M. I. Sherman, Y. B. Samchynska, V. M. Kobets, Development of an electronic system for remote assessment of students' knowledge in cloud-based learning environment, CTE Workshop Proceedings 9 (2022) 290–305. doi:10.55056/cte.121.

[17] A. V. Riabko, T. A. Vakaliuk, Physics on autopilot: exploring the use of an AI assistant for independent problem-solving practice, Educational Technology Quarterly 2024 (2024) 56–75. doi:10.55056/etq.671.

[18] T. V. Shabelnyk, S. V. Krivenko, N. Y. Rotanova, O. F. Diachenko, I. B. Tymofieieva, A. E. Kiv, Integration of chatbots into the system of professional training of masters, CTE Workshop Proceedings 8 (2021) 212–220. doi:10.55056/cte.233.

[19] V. A. Hamaniuk, The potential of Large Language Models in language education, Educational Dimension 5 (2021) 208–210. doi:10.31812/ed.650.