معامل التأثير العربي 2.03 لسنة 2023	مجـلة الـتــربـوي Journal of Educational	
العدد 25 ــ يوليو 2024	ISSN: 2011- 421X Arcif Q3	

مجلة التربوي

مجلة علمية محكمة تصدر عن كلية التربية الخمس

جامعة المرقب

العدد الخامس والعشرون

يوليو 2024م

- **المجلة ترحب بما يرد عليها من أبحاث وعلى استعداد لنشرها بعد التحكيم .**
- **المجلة تحترم كل الاحترام آراء المحكمين وتعمل بمقتضاها .**
- **كافة الآراء والأفكار المنشورة تعبر عن آراء أصحابها ولا تتحمل المجلة تبعاتها .**
- **يتحمل الباحث مسؤولية الأمانة العلمية وهو المسؤول عما ينشر له .**
- **البحوث المقدمة للنشر لا ترد لأصحابها نشرت أو لم تنشر .**

معامل التأثير العربي 2.03 لسنة 2023	مجـلة الـتــربـوي Journal of Educational	
العدد 25 ــ يوليو 2024	ISSN: 2011- 421X Arcif Q3	

ضوابط النشر:

يشترط في البحوث العلمية المقدمة للنشر أن يراعى فيها ما يأتي :

- أصول البحث العلمي وقواعده .
- ألا تكون المادة العلمية قد سبق نشرها أو كانت جزءا من رسالة علمية .
- يرفق بالبحث تزكية لغوية وفق أنموذج معد .
- تعدل البحوث المقبولة وتصحح وفق ما يراه المحكمون .
- التزام الباحث بالضوابط التي وضعتها المجلة من عدد الصفحات ، ونوع الخط ورقمه ، والفترات الزمنية الممنوحة للتعديل ، وما يستجد من ضوابط تضعها المجلة مستقبلا .

تنبيهات :

- للمجلة الحق في تعديل البحث أو طلب تعديله أو رفضه .
- يخضع البحث في النشر لأولويات المجلة وسياستها .
- البحوث المنشورة تعبر عن وجهة نظر أصحابها ، ولا تعبر عن وجهة نظر المجلة .

Information for authors

1- Authors of the articles being accepted are required to respect the regulations and the rules of the scientific research.
2- The research articles or manuscripts should be original and have not been published previously. Materials that are currently being considered by another journal or are a part of scientific dissertation are requested not to be submitted.
3- The research articles should be approved by a linguistic reviewer.
4- All research articles in the journal undergo rigorous peer review based on initial editor screening.
5- All authors are requested to follow the regulations of publication in the template paper prepared by the editorial board of the journal.

Attentions

1- The editor reserves the right to make any necessary changes in the papers, or request the author to do so, or reject the paper submitted.
2- The research articles undergo to the policy of the editorial board regarding the priority of publication.
3- The published articles represent only the authors' viewpoints.

* * *

مجـلة الـتــربـوي
Journal of Educational
ISSN: 2011- 421X
Arcif Q3
معامل التأثير العربي 2.03 لسنة 2023
العدد 25 ــ يوليو 2024

Investigating for: Is ChatGPT suitable for use as a teaching tool in a higher education to learning programming?

Salem Husein Almadhun[1], Aimen Abdalsalam Kleep[2], Osamah Alloush[3], Aimen M. Rmis[4], Ramadan Faraj Swese[5]

[1]Department of Computer, Faculty of Education, Elmergib University, Alkhums, Libya; salem.almadhun@elmergib.edu.ly
[2] Department of Computer, Faculty of Education, Elmergib University, Al Khums, Libya;
aakleeb@elmergib.edu.ly

[3] Department of Computer Science, Faculty of Science, Alasmarya Islamic University, Zliten, Libya; o.alloush@asmarya.edu.ly

[4] Department of Computer Science, Faculty of Science, Alasmarya Islamic University, Zliten, Libya; a.rmis@asmarya.edu.ly

[5] Department of Computer Science, Faculty of Science, Alasmarya Islamic University, Zliten, Libya; ramadanswese@asmarya.edu.ly

Abstract:

Launched in November 2022 by OpenAI.com, ChatGPT is an interactive chatbot that utilizes natural language processing (NLP) to interpret user inputs and generate appropriate responses (Ruby, 2023). Its introduction sparked interest in numerous fields, particularly within the realm of higher education. While academic dishonesty associated with ChatGPT has been a major concern in the academic world, this paper aims to highlight its potential as a pedagogical tool for instructing management information systems students in programming.

The background and development of ChatGPT are the main topics of this paper's succinct literature review. It provides an overview of ChatGPT's operation and an in-depth manual on the conversational modeling method (Ruby, 2023). An comprehensible explanation of large language modeling is provided for management information systems students. After a thorough analysis of ChatGPT and its features, the study plans and conducts an experiment to gauge how useful it is as a helping hand for novice

programmers working on a simple programming assignment.

The experiment aims to see if a new programming student can use ChatGPT's responses to complete the assignment. The experiment uses conversational prompts to describe the steps in the coding assignment and has ChatGPT return responses in the form of Python code blocks.

The paper describes each step in the process, including the prompts written and responses received from ChatGPT. This experimental assignment is to create a console application that will take input from the end user in the form of the user's first name, last name, and year born. The input feature of the program must validate each piece of end-user data. This validation will be performed in conditional while loops. The program will then write the end-user data into a tuple that will also be stored in a list. An outer conditional while loop should terminate when the index reaches five (5) total tuples with individual end-user data in the tuples. Once all the data is entered, the program will process the data into unique usernames and evaluate the usernames for duplicates.

The experiment's initial findings will establish if ChatGPT can effectively be used as a teaching aid to help students finish their programming projects. The usefulness and reproducibility of the prompts created for ChatGPT and the responses obtained from it will be assessed.

Keywords: AI, ChatGPT, Coding, Education.

Literature Review

Artificial intelligence text generators have sparked debates among professors regarding their use in academia. Some educators worry that students may utilize these tools to complete assignments independently, while others adopt a more open-minded view and see them as having the potential to accelerate learning. For instance, Anthony Picciano, an education professor at Hunter College, is less concerned about AI technologies and believes they could serve as valuable classroom tools if used to help students formulate ideas. (Aumann, A. 2023).

However, ChatGPT's capabilities may exceed initial expectations. In the research paper "ChatGPT User Experience: Implications for Education", Xiaoming Zhai emphasizes the need to acknowledge both the benefits and challenges presented by ChatGPT, as well as identify the core skillsets students must continue developing to thrive in the future. A study was conducted to gauge user experiences with ChatGPT, finding it easy to interact with and capable of human-like responses but also exposing reliability and

accuracy concerns. This highlights the importance of considering an AI tool's full implications for education. (Zhai, 2022).

Despite these reservations, Beth McMurtrie contends that ChatGPT and related resources are effective teaching resources. She views them as necessary, but she also recognizes that pupils must learn how to use them appropriately, considering both the advantages and disadvantages of the tools (McMurtrie, B. 2023).

Overall, the research points to ChatGPT as a useful tool for creating code examples and providing tailored feedback and advice in introductory programming courses offered in higher education. But, it's important to use caution and be conscious of ChatGPT's limitations when utilizing it as a stand-alone teaching tool.

Introduction

An artificial intelligence (AI) chatbot model called ChatGPT uses natural language processing to provide text responses to queries from people. In response to a human query, it can also produce programming code that functions. It is capable of comprehending spoken language and reacting to cues. With respect to its functionalities, ChatGPT can be used for a number of activities, including language translation, chatbot services, and automated customer support.

A branch of artificial intelligence called natural language processing, or NLP, makes it possible for computers to comprehend human speech in its natural environment. NLP seamlessly bridges the communication gap between complex human language and coded technology. In a bot-assisted environment, natural language processing (NLP) can be utilized to interpret user input. Imagine if chatbots responded to queries like people did and told jokes. This is a feature that developers are attempting to add to the chatbot, and NLP development is the only way to make it happen (Devlin et al., 2019).

Early natural language processing systems were based on n-gram models, which represent the probability of a word given the previous n-1 words in a sequence. These n-gram models found widespread use in applications like speech recognition and machine translation during the 1980s and 1990s (Jelinek, 1990). In the early 2000s, neural network approaches to language modeling were developed that demonstrated improved performance over n-gram models on various tasks. (Bengio et al., 2003)

There has been rapid progress in the size and complexity of language models over time, culminating in recent massive models such as GPT-3 which contains over 175 billion parameters. (Brown et al., 2020) In 2015, several technology leaders came together to

form OpenAI, a nonprofit focused on developing AI in a way that benefits humanity. OpenAI later launched The Gym in 2016 to allow reinforcement learning researchers to develop and benchmark their systems. (Kay, 2023)

In March of 2016, Microsoft debuted Tay, an AI chatbot designed for conversational interactions on Twitter and direct messages. Later that year, in December, Anthropic released Zo as a chatbot accessible through the Kik messenger app. These systems represented early commercial attempts to build conversational agents targeted at broader user bases.

Massive training data sets are necessary for large language models, however obtaining these sets of data might be difficult for various languages and topics. Significant computing resources, including specialist hardware like graphics processing units (GPUs) and tensor processing units (TPUs), are needed for training and using big language models. The tendency of large language models to overfit the training set makes it difficult to generalize the model to new tasks or data. Malicious uses of large language models include the creation of deepfakes and fake news. Large language models require a lot of energy to train, which might be bad for the environment.

How is ChatGPT operated?

Starting with a description from ChatGPT, this section of the paper will review, at a high level, how ChatGPT functions.

ChatGPT states that it is an NLP (natural language processing) model created by OpenAI. The GPT-3 model, which was trained on a vast corpus of text data, serves as its foundation. The model's goal is to produce natural language responses from user inputs that are human-like.

User input is converted into a vector representation by the ChatGPT model. After that, the GPT-3 model receives this vector representation and uses its vast corpus of text data to produce a response. The user is then given their response back after it has been decoded back into normal language.
A wide range of conversational subjects, such as sports, politics, music, movies, and more, were used to train the ChatGPT model. As a result, it can produce responses that are more precise and catered to the context and interests of the user. More training data can also be added to the model to further tailor it for certain use cases or domains.
Users can communicate with the model in real time through ChatGPT's interactive mode. In this mode, users can assist in further training the model by offering feedback

on the generated responses. This makes it more likely that the answers will be correct and pertinent to the discussion.
The GPT-3 language model from OpenAI, which was trained on a sizable corpus of online text, powers ChatGPT. This collection consists of content from books, internet, and other sources from the internet."

The acronym for ChatGPT is generative pre-trained transformer. Pre-training is the process of training the model on a broad corpus of text data before fine-tuning it for certain downstream tasks. In GPT, generative refers to the ability to generate new text based on the input it gets. Lastly, a kind of neural network design called the transformer is employed for tasks related to language modeling and natural language processing. Version 1 of OpenAI's Generative Pre-training Transformers, or GPT models, was introduced in 2018. Following that release, the models underwent rapid evolution, with the release of GPT-2 in 2019 and GPT-3 in 2020. The most recent version, known as InstructGPT and ChatGPT, was made available in 2022. With this release in 2022, supervised learning was used to incorporate user feedback into the system. The abundance of available content on the internet, lower processing costs, and improved computer performance have all contributed to the rapid evolution of the GPT model. Each of these occasions was beneficial.
The development of GPT-3 marked a significant advancement in large language models. Compared to its predecessor GPT-2, GPT-3 was trained on a significantly larger and more diverse dataset, allowing it to acquire a broader knowledge base capable of handling a more comprehensive range of tasks (Ruby, 2023).

ChatGPT is another large language model that leverages natural language processing and deep learning techniques, particularly the transformer architecture, to generate coherent and relevant human-like responses to user prompts (Meyer, 2023). Like other large language models, ChatGPT's capabilities are enabled by the rapid growth in computing power, which has facilitated the use of larger input datasets and model parameters to improve natural language understanding and generation (Ruby, 2023).

The increasing sophistication of large language models like GPT-3 and ChatGPT can be attributed to the advancements in natural language processing, a branch of machine learning focused on processing and understanding human language. By training on vast amounts of textual data, these models are able to learn the complex relationships between words and generate human-like responses, making them powerful tools for a wide range of applications.

Despite the surge in computing power and the vast amount of online text data, creating

conversational AI systems using large language models remained a significant challenge. A key obstacle was generating relevant and coherent responses to human prompts, ensuring that the responses stayed on topic and made sense within the conversational context. Traditional language models struggled with this, but ChatGPT's specialized training and fine-tuning on a massive dataset of conversational text enabled it to overcome this hurdle (Kelk, 2022). By learning the patterns and nuances of natural conversation, ChatGPT generates responses that contribute meaningfully to the dialogue.

Traditional language models often rely on predicting the next word in a sequence, typically using a Long Short Term Memory (LSTM) model to fill in the blanks with the most statistically probable word based on the prompt's context. This sequential modeling approach has two major limitations. First, the model struggles to accurately rank the likelihood of different words. Second, it processes the prompt as individual words in sequence rather than as a whole corpus, leading to a fixed context that extends beyond the original prompt by several steps. This restricts the complexity of relationships between words and the meanings that can be derived, as noted by Ruby (2023).

Generators were created by Google Brain in 2017 in order to get around these restrictions. Instead than processing data sequentially, these Generators could handle a whole prompt simultaneously. The GPT-3 and above big language models employ generator technology. This approach enables large amounts of computations to be performed in parallel and is utilized in neural network training. Large language model training times are greatly accelerated as a result. Pre-trained models are now able to handle ever-larger datasets with improved performance and more human-like replies because to these speed improvements in processing. Natural language processing has been deeply influenced by the generators, but others fear that as the model gets bigger and bigger, it may become monopolized. In 2021, Luitse and Denkena ChatGPT employ self-attention as a technique in addition to generator technology, which resulted in the development of the multi-head attention model.

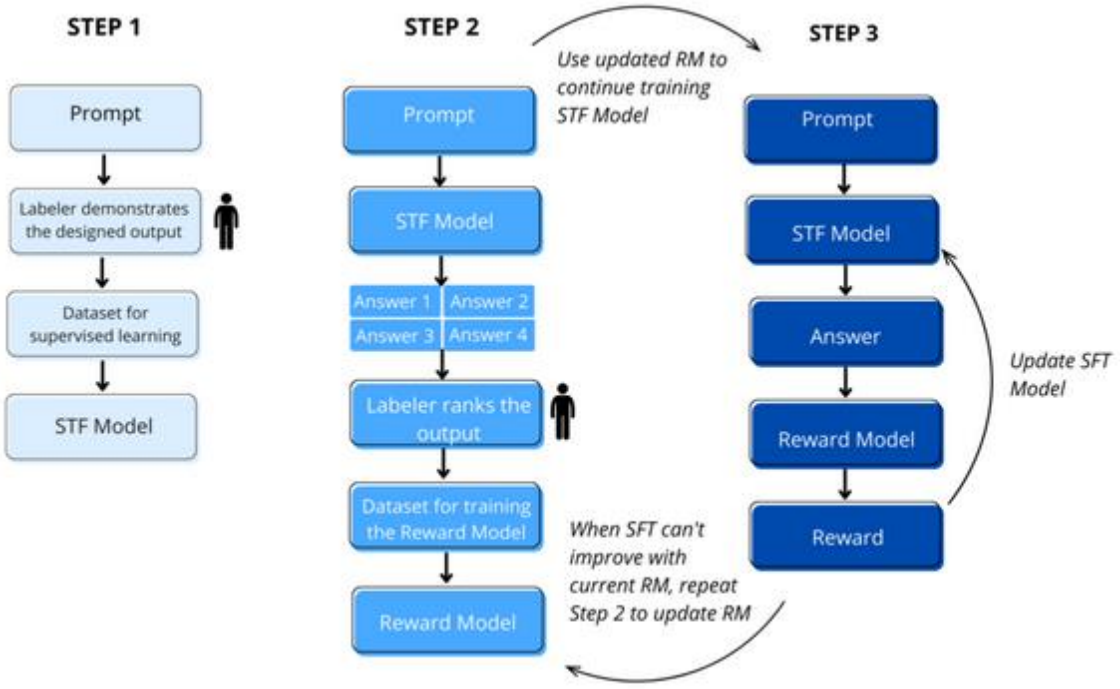
Luitse & Denkena (2021) state that in order to calculate the representation of ChatGPT's inputs (prompts) and outputs (responses), the generator model depends on an attention mechanism. When processing the text prediction task, the conventional attention-based architecture enables the model to concentrate on particular portions of the prompt. Nevertheless, it encodes the prompt as a series of memory vectors rather than attempting to compress the complete prompt into a fixed-length vector. This makes it possible to train the model by maximizing the information present in these memory vectors, enabling

the mode to concentrate only on the vectors that, when a prediction is being made, contain pertinent information. Longer sequences from a prompt can be processed more quickly when attention is used to discern which elements of the question are crucial and which are not. Through a technique known as multi-head attention, the generator makes it possible to calculate these attention vectors for more than one word at once. Encoding can be performed in parallel processing, acting on all the words in a prompt at once, but decoding can continue to be performed sequentially, producing the final answer one word at a time, with the use of a multi-head attention module. Since the generator employs more parallel processing during training, larger datasets may be employed. Furthermore, training times can be shortened by spreading out the processing burden over bigger clusters of processing units through increased usage of parallel computing.

The generator processes the input prompt and outputs the answer by means of encoding and decoding. One word at a time, the answer is generated by layers of nodes analyzing the input; each word is decided by analyzing the neural network analysis of the words that came before it in the sequence. There are two steps involved in creating a response. The input is first encoded, which converts it into a numerical format that the neural network can comprehend. This is the tokenization process, which converts each word into a token numerical representation for the neural network to use as input. The tokenization process then outputs the probability of the segment's next word. The next word in the segment is then determined using this token, and the process is repeated until the token has the highest likelihood of being the correct word. The second phase is decoding, which takes the output of a neural network and transforms it back into text that can be read by humans.

ChatGPT was taught employing a method known as Reinforced Learning with Human Feedback (RLHF), in addition to leveraging generator technology. The supervised learning procedure comprised three phases in this case. In the first phase, 40 contractors developed a supervised training data set using supervised fine-tuning (SFT), where the inputs (prompts) lead to predetermined and known outputs (responses). The responses produced in step 1 are scored in the second phase of implementing a reward model, and the model aims to maximize a scalar value as the output.

The model responds to a random cue in the last stage by utilizing the policy it developed in the second step. In order to maximize the reward it got, this policy is based on the lessons learnt by the machine in step 2. That reward is then fed back into the model to aid in the evolution of this policy, depending on the prompt/response pairing. Meyer (2023) here is a picture that illustrates this procedure.



(Calin Cretu, Machine Learning Engineer 2023)

The ChatGPT programming instruction experiment:

As part of our Master of Management Information Systems program, we offer an introductory programming course. The course uses custom-designed projects created by the teacher; these are not your typical programming exercises like Hangman or Tic Tac Toe. To enter data for the programs, we utilize real-world examples such using data structures, input functions, and text files to create usernames and passwords. For this experiment, we simply copied and lightly edited the instructions from our third Python project from Blackboard and placed them into the ChatGPT interface.

Unfinished program that wouldn't meet the criteria of the assignment was what was anticipated from the experiment. This theory was supported by the fact that there were several prerequisites for finishing the program. In a matter of seconds, ChatGPT's response was received. The code provided an almost feasible approach. The program executed without any errors, although lacking some of the necessary output. It is important to note that we may have "primed the pump" before providing ChatGPT the project three instructions because this prompt was the result of several previous efforts at Python coding prompts. We are not sure if the outcomes of this experiment would remain the same if a student conducted it without any past prompts or answers because ChatGPT does seem to build on earlier prompts.

It should be noted that while the code was functional and met most of the assignment's requirements, we desired it to have a "while" condition that checked for the number of workers entered using the "len" function. Instead, a "for" loop with a "in range" function was employed by the ChatGPT software. The ChatGPT usernames did not provide the format we desired, which was the username in all lower case. To get a decent grade, we had to create a program with the right code and output, but we also needed to have commented code. We neglected to inform the first ChatGPT application that we wanted comments, therefore it was without them. We then asked ChatGPT to comment the code by copying it back into a prompt, which it did without any problems.

According to the experiment's findings, ChatGPT was able to code the whole assignment in a way that would have earned the student an "A" mark. As a result, we determined that including ChatGPT into our lesson plan would be necessary.

Recommendations for using ChatGPT

Some recommendations to permit ChatGPT usage in the below circumstances:

- Apply as a Helper for Coding.
- Apply to develop elementary logic lessons.
- Apply to help troubleshoot code.
- Provide imaginative contexts and allow students to develop programs in any way they see fit (avoiding plagiarism).
- Establish the assignment's parameters, such as the use of a while loop for validating and entering data.
- Ascertain all the data structures and logic necessary to earn a satisfactory mark, but delegate ultimate coding approaches to the student.

Conclusion

There are more uses for ChatGPT than only finishing coding tasks. One major unforeseen consequence of ChatGPT will be plagiarism. Students studying programming may be able to prevent its unapproved usage by assigning extremely difficult coding tasks, but this will lead to unmanageable grading. The "join them if you can't beat them" strategy could be a preferable choice.

One way to do this would be to design projects that specify the coding techniques, functions, or properties that need to be employed, but let students to code the tasks anyway they see fit.

Additionally, ChatGPT may be used as a debugger to assist students in debugging their written code. In a future when ChatGPT-style AI is commonplace, the emphasis on programming will need to change from the technical aspects of the craft to the artistic

side of it. As a result, instead of teaching grammar as our primary focus as programming teachers; we will instead be teaching critical thinking, enabling the student to create and plan software solutions to solve business challenges. After that, they will generate those solutions by assigning ChatGPT to handle the tedious code while they concentrate on putting the solutions into practice.

***References*:**

1) Aumann, A. (2023,January 17). Professors Are Scrambling as Students Turn in AI. New York Observer, The (NY), 1.

2) Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. Journal of Machine Learning Research, 3(Feb), 1137-1155.

3) Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

4) Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171-4186.

5) Jelinek, F. (1990). Self-organizing statistical models for speech recognition. Readings in speech recognition, 623-659.

6) Kay, G. (2023, Feb 1). The history of ChatGPT creator OpenAI, which Elon Musk helped found before parting ways and criticizing. Retrieved from https://www.businessinsider.com/history-of-openai-company-chatgpt-elon-musk- founded-2022-12.

7) Kelk, I. (2023, January 11). How does chatgpt work? Clarifai. Retrieved February 6, 2023, from https://www.clarifai.com/blog/how-does-chatgpt-work.

8) Luitse, D., & Denkena, W. (2021). The Great Transformer: Examining the role of large language models in the political economy of ai. Big Data & Society, 8(2), 205395172110477. https://doi.org/10.1177/20539517211047734.

9) McMurtrie, B. (2023). Teaching: Will ChatGPT Change the Way You Teach? The Chronicle of Higher Education, retrieved March 27, 2023 from https://www.chronicle.com/newsletter/teaching/2023-01-05?cid=gen_sign_in

10) Meyer, P. (2023, March 7). ChatGPT: How Does It Work Internally? - Towards AI. Medium. https://pub.towardsai.net/chatgpt-how-does-it-work-internally- e0b3e23601a1

11) Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730-27744.

12) Ruby, M. (2023, January 31). How CHATGPT works: The models behind the bot. Medium. Retrieved February 6, 2023, from https://towardsdatascience.com/how-chatgpt-works-the-models-behind-the-bot- 1ce5fca96286

13) Zhai, Xiaoming, ChatGPT User Experience: Implications for Education (December 27, 2022). Available at SSRN: https://ssrn.com/abstract=4312418 or http://dx.doi.org/10.2139/ssrn.4312418.

14) Khaleel, N. A., Almadhun, S. H., Khalil, H. A., Alasoud, A. M., & Rmis, A. M. (2023). The Impact Of Mobile Banking Services On Customer Satisfaction And The Factors That Influence Users'intention to Use Them.

15) Hanan A. Khalil, Aimen M. Rmis, Salem H. Almadhun , Tareg A. Elawaj, Walid F. Naamat. (2023). The creation of theoretical frameworks to establish sustainable adoption of e-health in Libya. Humanitarian and Natural Sciences Journal, 4(7). doi:10.53796/hnsj4716

16) Almadhun, Salem, Mehmet TOYCAN, and Ahmet ADALIER. "VB2ALGO: An Educational Reverse Engineering Tool to Enhance High School Students' Learning Capacities in Programming." Revista de Cercetare si Interventie Sociala 67 (2019): 67–87. https://doi.org/10.33788/rcis.67.5.

17) Almadhun, Salem Husein, Salem M Aldeep, Aimen   M Rmis, and Khairia A Amer. "Examination of 4G (LTE) Wireless Network." El tarbawe journal 19, no. 1 (July 2021): 285–94. https://doi.org/http://dspace.elmergib.edu.ly/xmlui/handle/123456789/1119.

18) Almadhun, S.H. et al. (2024) 'Programming Education in focus: Investigating and analyzing the present state in Libyan primary and Secondary Schools', Humanitarian and Natural Sciences Journal, 5(1), pp. 342–357. doi:https://doi.org/10.53796/hnsj412/22.

19) Almadhun, S.H. et al. (2023) 'Harnessing mobile phones as innovative teaching tools in Libyan Primary Schools', Humanitarian and Natural Sciences Journal, 4(12), pp. 342–357. doi:https://doi.org/10.53796/hnsj412/22.