

**CHATGPT: KONTROL EDİLEBİLİR METİN ÜRETİMİNİN YAZILIM
MÜHENDİSLİĞİ EĞİTİMİNE ETKİSİ**
**CHATGPT: IMPACT OF CONTROLLABLE TEXT GENERATION ON SOFTWARE
ENGINEERING EDUCATION**

Cansu Çiğdem Ekin
Atılım Üniversitesi
Bilgisayar Mühendisliği

Güzin Türkmen
Atılım Üniversitesi
Bilgisayar Mühendisliği

Murat Karakaya
Atılım Üniversitesi
Bilgisayar Mühendisliği

Özet: Kontrol Edilebilir Metin Oluşturma, verilen bir soru için Yapay Zeka tekniklerini kullanarak yanıtlar oluşturmaktır. Son dönemde Derin Öğrenme yaklaşımı yardımıyla Kontrol Edilebilir Metin Oluşturmayı gerçekleştirmek için geliştirilmiş çeşitli modeller önerilmiştir. En yakın olarak, ChatGPT adlı çok güçlü bir model bir web arayüzü üzerinden kamu ile paylaşılmıştır. ChatGPT, verilen bir soruya metin yanıtları üreten bir Derin Öğrenme Transformer modelidir. Kontrol Edilebilir Metin Oluşturma modellerinin güçlü bir örneği olarak ChatGPT ile çok sayıda testler gerçekleştirdik. ChatGPT gibi Kontrol Edilebilir Metin Oluşturma modellerinin, öğrenciler için daha etkileşimli ve kişiselleştirilmiş bir öğrenme deneyimi sağlayarak Yazılım Mühendisliği eğitiminde büyük etki yaratma potansiyeline sahip olduğuna inanıyoruz. Bununla birlikte, eğitimde Kontrol Edilebilir Metin Oluşturma modellerinin kullanılmasının getirdiği potansiyel riskler ve dezavantajlar da vardır. Bu makalede, Kontrol Edilebilir Metin Oluşturma modellerinin özellikle Yazılım Mühendisliği eğitiminde yaratacağı zorlukları ve fırsatları eğitimcilerin ve öğrencilerin bakış açısından ele alıyoruz. ChatGPT ile yapılan deneylerin sonuçlarını gözlemleyerek, eğitimciler ve öğrenciler için Yazılım Mühendisliği eğitiminde yapay zekaya dayanan bir metin oluşturucunun potansiyellerini ve risklerini sunuyor ve tartışıyoruz.

Anahtar Kelimeler: Yapay Zeka, Kontrol Edilebilir Metin Oluşturma, ChatGPT, Yazılım Mühendisliği, Eğitim

Abstract: Controllable Text Generation (CTG) is a method that uses Artificial Intelligence (AI) techniques to generate responses for text input. In recent years, with the help of Deep Learning (DL) approach, there have been various transformer models developed for realizing CTG. Recently a very powerful model, named ChatGPT, has been shared with the community via a web interface. ChatGPT is a DL Transformer model that generates text replies to a provided question. We conduct a number of tests with ChatGPT as a powerful example of CTG models. We believe that CTG models like ChatGPT have the potential to impact Software Engineering education by

providing a more interactive and personalized learning experience for students. However, there are also potential risks and drawbacks introduced by using CTG models in education. In this article, we walk through the challenges and opportunities that CTG models would create, especially in software engineering education, from the point of view of educators and students. Observing the results of the experiments with ChatGPT, we present and discuss the potentials and risks of an artificial intelligent text generator in Software Engineering education for educators and students.

Keywords: Artificial Intelligence, Controllable Text Generation, ChatGPT, Software Engineering, Education

Introduction

In recent years, we have witnessed groundbreaking innovations in Natural Language Processing (NLP) with the help of Transformer Deep Learning models. Many novel tasks have been accomplished by a variety of these models. One of them is Controllable Text Generation (CTG). CTG is a task in natural language processing (NLP) that involves generating text that adheres to certain constraints or meets specific objectives. These constraints or objectives can include things like style, tone, length, or content. In recent years, transformers have become a popular choice for CTG due to their ability to process large amounts of data and generate high-quality text.

There have been numerous transformer models developed for CTG in recent years. Some of the most popular ones include:

- GPT-3 (Generative Pre-trained Transformer 3) is a large-scale transformer model developed by OpenAI [1]. It is trained on a massive dataset of web pages and has been shown to perform well on a wide range of NLP tasks, including CTG.
- CTRL (Controllable Transformer) is a transformer model developed by Salesforce that is designed specifically for CTG [2]. It is trained using a combination of supervised and unsupervised learning and can generate text that is coherent and stylistically consistent with a given prompt.
- T5 (Text-To-Text Transfer Transformer) is another large-scale transformer model developed by Google that is designed for a wide range of NLP tasks, including CTG [3]. It is trained on a large dataset of web pages and has been shown to generate high-quality text that is coherent and stylistically consistent with a given prompt.
- TextGAN (Generative Adversarial Networks for Text) is a transformer model developed for CTG that utilizes the principles of generative adversarial networks (GANs) [4]. It can generate text that is coherent and stylistically consistent with a given prompt while also being diverse and creative.

The interested reader can refer to these surveys to learn more about CTG models [5, 6]. Overall, these transformer models have shown promising results in CTG and have the potential to revolutionize the field of NLP. However, there is still much room for improvement and further research is needed to fully realize the potential of these models.

In this paper, we would like to demonstrate the possible use cases of CTG models and discuss their impact on Software Engineering Education for educators and learners. We believe that CTG has the potential to greatly impact software engineering education by improving various

processes and tasks. These include generating code documentation, automating code review comments, increasing productivity, enhancing communication, providing personalized learning materials, increasing efficiency, etc. On the other hand, it can cause several problems such as plagiarism, un-factual arguments, etc. Overall, CTG has the potential and risks to software engineering education for both students and instructors.

In this paper, we will use the recently published ChatGPT transformer model to demonstrate these potentials and risks. According to OpenAI, “*ChatGPT can interact conversationally. The dialogue format makes it possible for ChatGPT to answer follow-up questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests*”. The reader can access and interact with the ChatGPT transformer from the [OpenAI](#) website [7].

Potentials for Learners and Educators

In the context of software engineering education, CTG can have several positive impacts on behalf of educators and learners.

Personalized Learning Materials

One of the major advantages of using CTG for educators is the ability to create personalized teaching materials for their students. By using CTG, educators can generate instructional materials that are tailored to the specific needs and learning styles of their students, which can be more effective than using generic materials. This can be particularly useful for students who may have learning disabilities or other challenges that require more customized instructional approaches.

Considering the benefits of using ChatGPT as personalized educational material for students; CTG can generate personalized learning materials such as tutorials and exercises based on a student's individual needs and learning style, which can help students learn more effectively and efficiently [8]. For example, an educator could input a list of topics that a student needs to learn, along with any specific requirements or preferences, and ChatGPT could generate a personalized lesson plan or quiz based on this information.

Personalized Feedback and Guidance

Another advantage of CTG for educators is the ability to provide personalized feedback and guidance to students as they work on programming assignments. By doing so, educators can save time by automating the process of providing feedback and guidance to students. This can allow educators to focus on more high-level tasks, such as helping students with more complex concepts or providing one-on-one support. For example, an instructor or mentor could input a student's code, along with any specific feedback or guidance they would like to provide, and ChatGPT could generate personalized feedback based on this information. This could be particularly useful for providing individualized feedback to students in a large class or online course, as it would allow the instructor or mentor to provide more detailed and personalized feedback without having to spend as much time reviewing each student's work individually.

On behalf of students, personalized feedback help students better understand their strengths and weaknesses and can provide them with targeted support and guidance to improve

their performance and learning outcomes. Students may be more likely to understand and retain information when it is presented in a way that is tailored to their needs and learning style.

Automatic Code Generation and Testing

One of the main advantages of using CGT to generate code is that it can save students a significant amount of time (see Figure 1). This is especially useful for students who are working on large, complex projects or assignments where writing code from scratch can be time-consuming. Generating code automatically also helps to reduce the number of errors in a program. By generating code automatically, students can focus on higher-level tasks, such as designing algorithms or testing code, rather than spending time on the tedious process of writing code. This can improve productivity and allow students to make more progress on their projects and a learning opportunity for students. By examining the generated code, students can learn about coding conventions and best practices, and they can use this knowledge to improve their coding skills. CTG can also be used to generate code automatically based on certain constraints or requirements, allowing students to practice their coding skills and receive immediate feedback on their work. Thereby improving their coding skills more quickly and effectively [9, 10, 11]. In Figure 2, a code with errors is entered into ChatGPT, and ChatGPT responds to the user with a corrected version of the code and an explanation.

CTG can also be useful for software engineering educators in terms of generating code automatically. Some advantages include it can save educators a significant amount of time. It can help improve the efficiency of the educational process. For example, educators can use CTG to generate code for use in classroom demonstrations or to create code samples for students to work with. CTG can also help ensure that code is consistent across different projects or assignments. This can be useful for educators who want to ensure that students are being held to the same standards and that they are receiving a consistent learning experience.

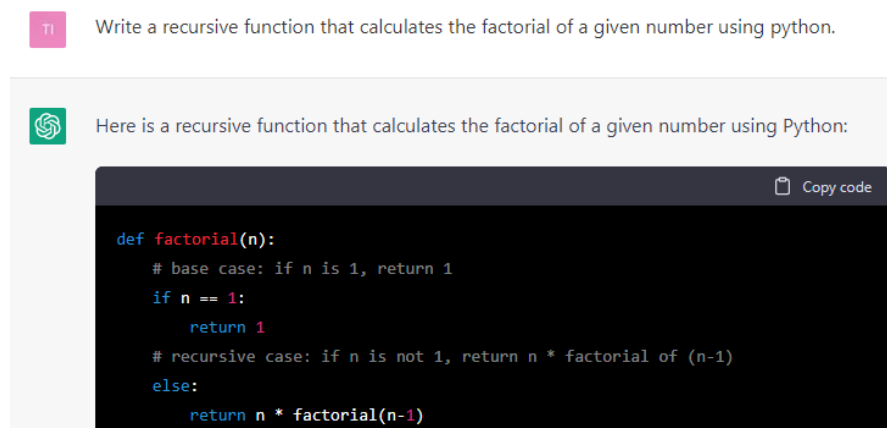


Figure 1. Code generated with ChatGPT

On the other hand, CTG has potential benefits for software testing. Some of these benefits include:

- Automation: ChatGPT can automate certain tasks associated with software testing, such as generating test cases and test data. This can save educators time and resources (See Figure 2).

- Scalability: ChatGPT can be used to test large and complex software systems, which may be difficult for educators to do manually.
-
- Flexibility: ChatGPT can be used for various types of software testing, such as functional, regression, and stress testing.



write a test function for the function above



Here is an example test function that can be used to test the `equalDigits` function:



Copy code

```
#include <stdio.h>
#include <assert.h>

void equalDigits(int num);

void test_equalDigits() {
    // Test case 1: check last two digits are equal
    int input = 1233;
    equalDigits(input);
    assert(sizeof("Last two digits are equal\n"));

    // Test case 2: check last two digits are not equal
    input = 1234;
    equalDigits(input);
    assert(sizeof("Last two digits are not equal\n"));

    // Test case 3: check last two digits are equal when input is negative
    input = -1233;
    equalDigits(input);
    assert(sizeof("Last two digits are equal\n"));

    printf("All test cases passed\n");
}

int main() {
    test_equalDigits();
}
```

Figure 2 Screenshot of code testing with ChatGPT

Collaborative Learning

In collaborative learning environments, CTG can facilitate the sharing of ideas and the generation of code, creating a more interactive and engaging learning experience. Additionally, it can help students improve their communication skills by generating clear and accurate documentation and reports, which can be especially useful for students who are not native English speakers. Also, it can help ensure that the content provided to students is consistent and accurate.

This can be especially important in collaborative learning environments, where multiple educators may be working together to deliver content to students.

Considering the benefits to educators, CTG can facilitate the sharing of ideas and the generation of code in collaborative learning environments, creating a more interactive and engaging learning experience. CTG can allow educators to generate customized, relevant, and accurate content for their students, which can help engage and motivate them to learn.

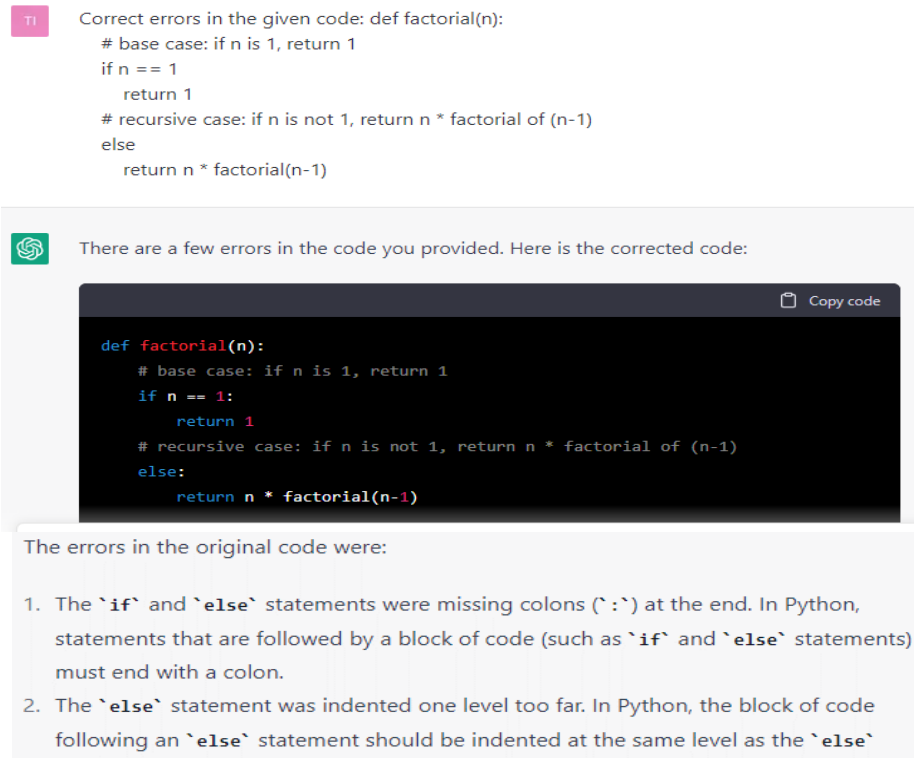


Figure 3. Code correction and a part of the explanation with ChatGPT

Generating code documentation

Code comments, commit messages, and release notes can improve a developer's understanding of a program by providing clear and thorough documentation [12, 13]. CTG can also generate high-quality documentation for code, which can be an important resource for developers and help students better understand the purpose and functionality of the code, as well as facilitate collaboration within a team. Three of the most widely used applications that operate within these frameworks are generating code from natural language descriptions, generating documentation, and automatically repairing programs [14].

CTG can save time for educators and mentors by automating the process of creating code documentation as well. This can be especially useful for educators working with large classes or online courses, where the volume of code to be reviewed and documented can be overwhelming. A python code documentation example using ChatGPT is given in Figure 4 and 5 respectively. The user gives the code to be documented in Figure 4 then, ChatGPT gives a documented version of the code in Figure 5. It is important to note that while CTG can be a useful tool for improving the efficiency and effectiveness of software engineering education, it is not a replacement for human instruction and guidance.

```

#include <stdio.h>

// define a structure for a student
struct student {
    char name[50];
    int age;
    char grade;
};

int main() {
    // define a variable of type 'struct student'
    struct student s;

    // assign values to the fields of the structure
    strcpy(s.name, "John");

```

Figure 4. A part of a code to be documented

Here is a more detailed version of the documentation for the code you provided:

Copy code

```

#include <stdio.h>

/*
 * This program defines a structure called 'student' that represents a student
 * with three fields:
 * - name: a string of up to 50 characters that represents the student's name
 * - age: an integer that represents the student's age
 * - grade: a character that represents the student's grade
 */
struct student {
    char name[50];
    int age;
    char grade;
};

int main() {
    // define a variable of type 'struct student'

```

Figure 5. A part of a documented code from ChatGPT

Risks for Learners and Educators

There are several risks associated with using ChatGPT or similar language models in education. These risks include:

Plagiarism

ChatGPT may generate original text, but it is important to recognize that this text is not the work of the student or educator using the tool. Using ChatGPT or similar language models to generate assignments (writing code, essay about a topic) or other written work could be considered plagiarism [15, 18, 26]. A sample screenshot is given from ChatGPT on the C programming question in Figure 5. The related application can easily answer any question about programming in many languages in an error-free and unique way. Therefore, the possibility of

plagiarism in student homework, especially in programming courses, requires new measurement and evaluation methods to be considered.

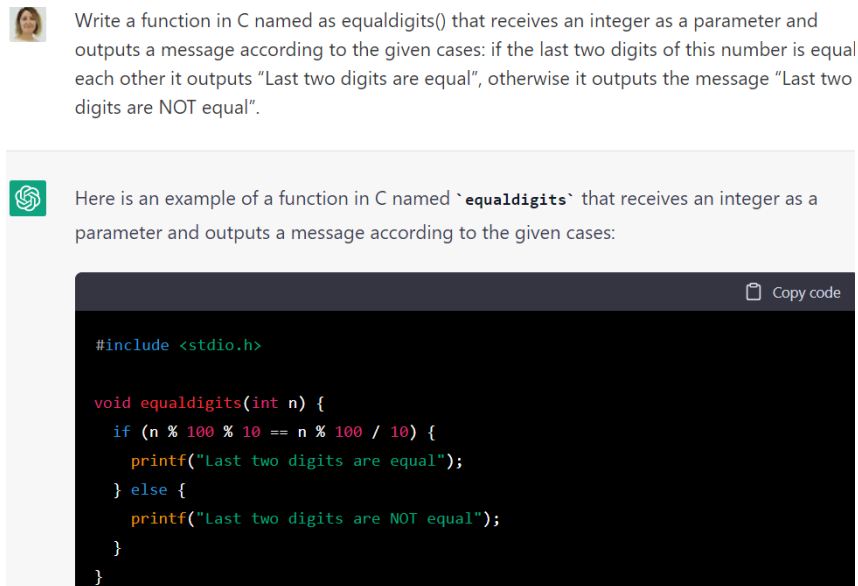


Figure 6 Screenshot of code writing with ChatGPT

Dependence

For the software engineering field, this application can assist with debugging, testing, troubleshooting and to provide guidance and suggestions for code improvements. Educators and learners should be mindful of their reliance on ChatGPT and ensure that they are using it as a supplement to, rather than a replacement for, traditional teaching and learning methods. While these models can help generate ideas or provide inspiration, it is important for students to develop their writing skills and take responsibility for the content they produce.

Relying too heavily on ChatGPT or similar models could lead to a lack of critical thinking and originality, which are important skills to develop in an educational setting. On the other hand, the possibilities offered by ChatGPT require a higher level re-evaluation of the free rider problem in education. The free rider can occur when individuals can access education resources or services without paying for them, or when they can benefit from the efforts of others without contributing to those efforts [19, 20].

Misinformation

The use of ChatGPT as a search engine like google is among the frequently discussed topics [15]. The corresponding application may provide a new experience for learners and educators of easily viewing information that exceeds the functionality of currently available search engines, by selecting the most relevant material related to the search query and providing clear explanations for it. But it should not be forgotten that ChatGPT is trained on a large dataset of text from the internet, and this dataset may contain misinformation or biased information.

As a result, ChatGPT may provide incorrect or misleading answers to questions. ChatGPT, which is pretty good at coding, can make mistakes in basic algebra problems. He also doesn't seem very good at counting. When asked how many letters are in the word sixteen, he can

give the wrong answer (see Figure 7). It is important for educators and learners to fact-check any information provided by ChatGPT and to critically evaluate its sources [16, 17].

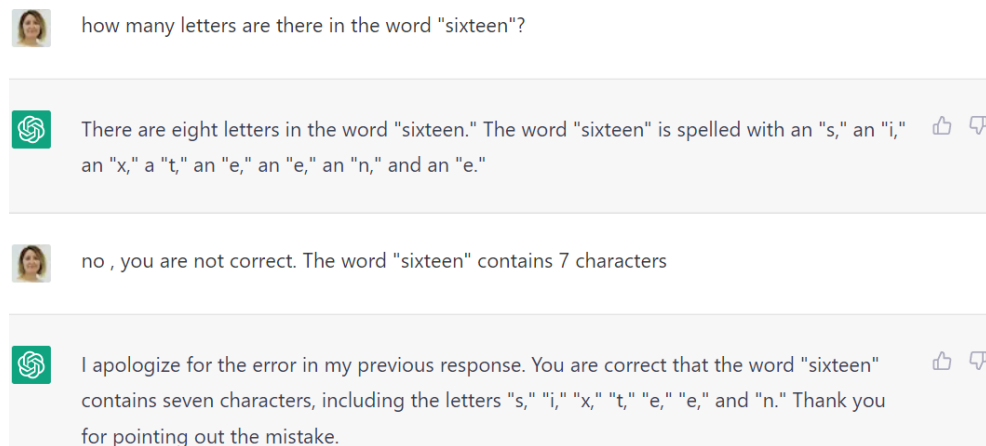


Figure 7. Screenshot of ChatGPT Wrong Answer

Distraction

ChatGPT can be a fun and engaging tool for learners (see Figure 7), but it can also be a distraction if not used appropriately. The distraction effect of technology on students' concentration refers to the ability of technology, particularly digital devices and tools, to divert students' attention away from their studies and other educational activities [21, 22] Especially if educators plan to use ChatGPT as a supportive tool for lesson activities in the classroom environment, they should consider the distracter effect of the relevant technology. Students may become more focused on interacting with the ChatGPT and generating responses than on the educational content being presented.

For example, if an educator is using ChatGPT to supplement a lesson on a particular topic, students may be more interested in chatting with the model and less interested in paying attention to the lesson. This could lead to a decline in student engagement and potentially hinder their understanding of the material. The potential undesirable impacts of different technologies on students' focus in college classrooms have been extensively discussed in recent literature [22, 23, 24]. This has caused some professors and colleges to forbid the use of electronic devices (such as laptops) in the classroom [25]. Fried investigated the impact of laptop use in the classroom on student learning and discovered that it had a damaging effect on both performance and learning [24].

Ethical Considerations

There are also ethical considerations to be aware of when using ChatGPT and similar language models. For example, ChatGPT is a machine learning model that has been trained on a dataset of text from the internet, which may include sensitive or personal information. For example, if ChatGPT is used to generate written work (see Figure 8), it may draw upon sensitive or personal information that it has been trained on, which could raise privacy concerns. Additionally, there may be ethical implications to consider, such as the potential for ChatGPT to spread misinformation or to be used as a tool for plagiarism. It is important to consider the privacy and ethical implications of using such a tool in education [17].

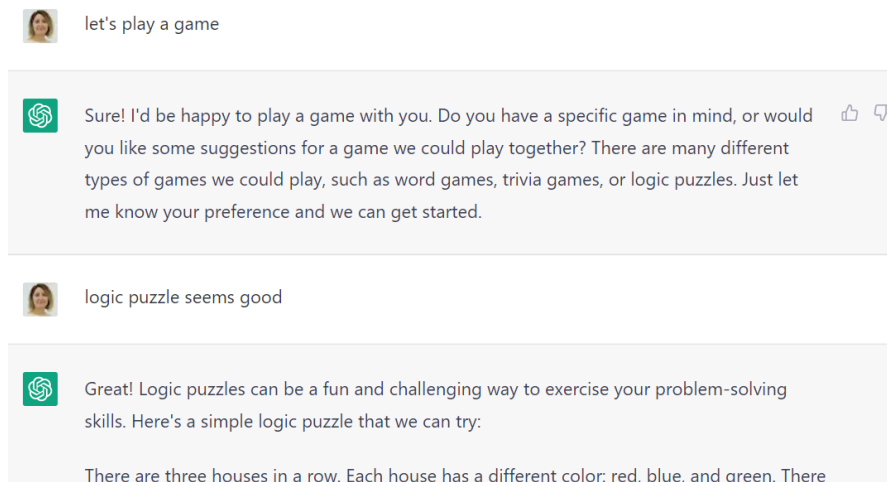


Figure 8. Screenshot of Using ChatGPT as a Game Tool

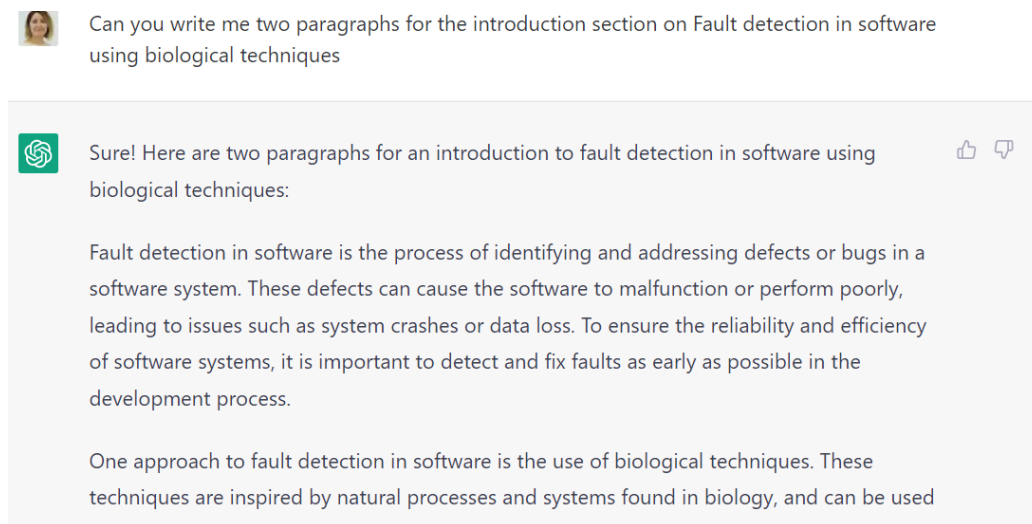


Figure 9. Writing an Academic Essay with ChatGPT

Discussion and Conclusions

As discussed above using a powerful CTG model can provide new opportunities in Software Engineering education as well as introduce new challenges. In this section, we would like to discuss the possible directions and precautions that can be taken for adapting this new tool.

First, we believe that as educators we should be aware of these kinds of tools available for students and educators. We need to incorporate these kinds of tools into education materials and get prepared to use them and help students to benefit from them.

After admitting their existence, we need to review the education process including measurement and evaluation techniques. For example, grading homework and projects might not be a fair or wise option if an answer can be generated from this kind of tool. As we have experimented with ChatGPT, we observed that it was able to generate almost always correct answers for real exams and homework questions of various Programming Language (PL) courses including C, C++, Java, etc. We also tried other courses exams and homework questions

including operating systems, algorithms, data structures, machine learning, requirements engineering, software design, validation and verification, software process, etc. The generated text was most of the time valid and concise. Therefore, we believe that we should not grade homeworks and projects if they are out-of-class activities. We can allocate time during classes or labs for homeworks and projects under the instructor's monitor ensuring they are not using the CTG tools.

Using ChatGPT to generate written work, such as assignments or essays, could be considered plagiarism and it is important for students to develop their writing skills and take responsibility for the content they produce. As a large language model developed by OpenAI, ChatGPT can generate human-like text, which has made it a useful tool for tasks such as code completion and automated programming feedback. On the other hand, this feature of ChatGPT can de-motivate students to learn PLs. This is an important issue if we consider real-life work environments as well. Assume that in a company you are one of the software engineers and you are tasked to implement a work package. Instead of coding by yourself, you can use these tools and check the generated code and use it. This kind of tool may increase productivity drastically. In another scenario, if there is a bug in your code you can request help from this tool and fix the issue soon. However, during education, the availability of these kinds of tools can discourage a student to learn and excel in a PL. As educators, we should be ready to answer such questions: "Do we really need to teach programming?"

In software engineering education, ChatGPT has been used to provide students with personalized feedback on their code, as well as to generate practice problems and quizzes. This has helped to improve the efficiency and effectiveness of software engineering education, as students are able to receive timely and targeted feedback on their progress, and instructors can focus more on higher-level concepts and problem-solving skills.

However, it is important to note that ChatGPT is a machine learning model, which means it is only as good as the data it is trained on. If the training data is biased or incomplete, the model may produce biased or inaccurate output, which can have negative consequences for software engineering education. It is therefore important for educators to carefully evaluate the output of ChatGPT and to use it in combination with other teaching methods and resources.

Additionally, ChatGPT and other technological tools can be distracting if not used appropriately, and educators should consider the potential distraction effect in educational settings.

In conclusion, ChatGPT is a powerful tool that can provide numerous benefits to software engineering education, such as personalized feedback, practice problems, and code generation. However, it is important for educators to be aware of its limitations and potential biases, and to use it in combination with other teaching methods. Additionally, ChatGPT should not be used to generate written work, as this may be considered plagiarism. It is crucial for students to develop their writing skills and take responsibility for the content they produce. Educators should also be aware of the potential for ChatGPT to provide misinformation or biased information, and students should be encouraged to fact-check and critically evaluate its sources. Additionally, ChatGPT and other technological tools should be used appropriately to minimize distractions in the classroom. While ChatGPT can be a useful supplement to traditional teaching methods, it should not be relied upon as a replacement for human instruction and critical thinking skills. It is important to carefully consider the limitations and potential ethical issues associated with using CTG in education and to ensure that it is used responsibly and transparently.

References

- [1] Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.
- [2] Keskar, Nitish Shirish, et al. "Ctrl: A conditional transformer language model for controllable generation." *arXiv preprint arXiv:1909.05858* (2019).
- [3] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *J. Mach. Learn. Res.* 21.140 (2020): 1-67.
- [4] Zhang, Yizhe, et al. "Adversarial feature matching for text generation." *International Conference on Machine Learning*. PMLR, 2017.
- [5] Li, Junyi, et al. "Pretrained language models for text generation: A survey." *arXiv preprint arXiv:2105.10311* (2021).
- [6] de Rosa, Gustavo H., and João P. Papa. "A survey on text generation using generative adversarial networks." *Pattern Recognition* 119 (2021): 108098.
- [7] ChatGPT: Optimizing language models for dialogue. Accessed December 10, 2022. [Online]. Available: <https://openai.com/blog/ChatGPT/>
- [8] A. Mastropaolo et al., "Studying the Usage of Text-To-Text Transfer Transformer to Support Code-Related Tasks," 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 336-347, doi: 10.1109/ICSE43902.2021.00041.
- [9] K. Naidu. (2022) ChatGPT is a new ai chatbot that can find errors in your code and write you a story. Accessed December 10, 2022. [Online]. Available: <https://crazyengineer.in/ChatGPT-is-a-new-ai-chatbot/>
- [10] Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. A transformer-based approach for source code summarization, 2020.
- [11] Colin B. Clement, Dawn Drain, Jonathan Timcheck, Alexey Svyatkovskiy, and Neel Sundaresan. Pynt5: multi-mode translation of natural language and python code with transformers, 2020.
- [12] Andrew Forward and Timothy C Lethbridge. 2002. The relevance of software documentation, tools and technologies: a survey. In *Proceedings of the 2002 ACM symposium on Document engineering*. 26–33.
- [13] Xing Hu, Qiuyuan Chen, Haoye Wang, Xin Xia, David Lo, and Thomas Zimmermann. 2022. Correlating Automated and Human Evaluation of Code Documentation Generation Quality. *ACM Trans. Softw. Eng. Methodol.* 31, 4, Article 63 (July 2022), 28 pages. <https://doi.org/10.1145/3502853>
- [14] E. Dehaerne, B. Dey, S. Halder, S. De Gendt and W. Meert, "Code Generation Using Machine Learning: A Systematic Review," in *IEEE Access*, vol. 10, pp. 82434-82455, 2022, doi: 10.1109/ACCESS.2022.3196347.
- [15] Haque, M. U., Dharmadasa, I., Sworna, Z. T., Rajapakse, R. N., & Ahmad, H. (2022). "I think this is the most disruptive technology": Exploring Sentiments of ChatGPT Early Adopters using Twitter Data. *arXiv preprint arXiv:2212.05856*.
- [16] Yu, J., & Egger, R. Category Archives: Blog. Retrieved 18 December 2022, from <http://www.datascience-in-tourism.com/?cat=3>

- [17] Frye, B. L. (2022). Should Using an AI Text Generator to Produce Academic Writing Be Plagiarism?. *Fordham Intellectual Property, Media & Entertainment Law Journal*, Forthcoming.
- [18] Zhou, C., Qiu, C., & Acuna, D. E. (2022). Paraphrase Identification with Deep Learning: A Review of Datasets and Methods. *arXiv preprint arXiv:2212.06933*.
- [19] Ruel, G. and Bastiaans, N. (2003), "Free-riding and team performance in project education", *International Journal of Management Education*, Vol. 3 No. 1, pp. 26-37.
- [20] Strong, J. and Anderson, R. (1990), "Free riding in group projects: control mechanisms and preliminary data", *Journal of Marketing Education*, Vol. 12 No. 2, pp. 61-67.
- [21] Spitzer, M. (2014). Information technology in education: Risks and side effects. *Trends in Neuroscience and Education*, 3(3-4), 81-85.
- [22] Attia, N. A., Baig, L., Marzouk, Y. I., & Khan, A. (2017). The potential effect of technology and distractions on undergraduate students' concentration. *Pakistan journal of medical sciences*, 33(4), 860.
- [23] Shelton, J. T., Elliott, E. M., Eaves, S. D., & Exner, A. L. (2009). The distracting effects of a ringing cell phone: An investigation of the laboratory and the classroom setting. *Journal of environmental psychology*, 29(4), 513-521.
- [24] Fried, C. B. (2008). In-class laptop use and its effects on student learning. *Computers & education*, 50(3), 906-914.
- [25] Yamamoto, K. (2007). Banning laptops in the classroom: Is it worth the hassles?. *Journal of Legal Education*, 57(4), 477-520.
- [26] Francke, E., & Alexander, B. (2019, October). The potential influence of artificial intelligence on plagiarism a higher education perspective. In *Proc European Conference on the Impact of Artificial Intelligence and Robotics*. EM Normandie Business School, Oxford (pp. 131-140).