

SUMMARY

In this project, we have developed a chatbot, called SimpliCT. The purpose of SimpliCT is to act as a digital tutor, to assist students in learning Computational Thinking.

In the beginning, we started by researching Computational Thinking, and the part it plays in the educational system. We have found that Computational Thinking is a vital skill, that does not only find relevance in the fields of computer science. Computational Thinking is subconsciously used in our everyday life. Our research has found that children as early as in the third grade uses Computational Thinking to solve problems. Furthermore, we have researched how chatbots has been used in educational fields. Our research found both positive and negative effects of educational chatbots. We found multiple examples of chatbots used as learning tool. We found that chatbots can effect student engagement, make information easily available to students and even come up with solutions. We found a chatbot called VPbot, which had been used at the Harvard Medical School. A study found an significant increase in both test scores and learning efficiency for students who had used VPbot. Our research did not only find positive results. We found that the use of a chatbot could have a negative impact on problem solving and critical thinking for students. This is due to the ability for chatbots to directly provide the students with solutions, instead of encouraging them to solve the problem themselves. Furthermore, it is hard to verify the output given by a chatbot, as they have a tendency of confidently presenting information, even though it might be incorrect. The output seems convincingly real, due to the human-like language.

Even though we found that chatbots can have a negative impact on learning (when used incorrectly), we found that chatbots have great potential for assisting students in learning. We therefore wanted to develop a chatbot to assist students in learning Computational Thinking.

In the beginning of the development process for SimpliCT, we had to make a decision whether we wanted it to be a desktop- or a web application. Based on the fact that we wanted the bot to be easily available, we decided to develop SimpliCT as a web application. The application was written in JavaScript to facilitate web development, with the React library used to implement the frontend.

Before settling on an AI model, we investigated different ways for creating a chatbot; *Microsoft Bot Framework*¹, *ChatterBot*², and *OpenAI*³. We settled on using OpenAI, as they provide access to their GPT models. This allowed for us to implement our own user interface, without having to build or train a model from scratch, instead, letting us fine-tune the existing model to our needs.

SimpliCT uses the `gpt-3.5-turbo-0125` model, as this is the latest model to allow for fine-tuning. We fine-tuned the model with a dataset consisting of 68 training examples.

The current system is a prototype, consisting of a single client responsible for taking the users prompt, including it in an API request to OpenAI's servers, and displaying the response received from the GPT model.

We conducted a comparative study on students that had attended the Computational Thinking course at AAU last year, and people from our social circle without experience with Computational Thinking. Participants were recruited through convenience sampling. Both groups were asked to complete some tasks in order for them to gain familiarity with SimpliCT, after which they were presented with an Attrakdiff questionnaire. The Attrakdiff questionnaire evaluates four dimension of the user experience; *Pragmatic Quality*, *Hedonic Identification Quality*, *Hedonic Stimulation Quality* and *Attractiveness*.

Group A was presented with an exam set, in order for us to see if they had understood the information provided by SimpliCT. We graded the exam set on the Danish 7-point scale our selves. We assigned a max point value to each question, and awarded points to each answer, based on how fulfilling the answer was.

Group B was interviewed, to get an understanding of what they liked, disliked and their overall opinion on using SimpliCT to learn Computation Thinking. We only interviewed group B, as we considered that only they were able to compare learning Computational Thinking, with and without SimpliCT.

The findings of our study had let us to conclude that SimpliCT provide users with an acceptable user experience. We are not able to conclude that students are able to learn Computational Thinking from SimpliCT, as this would require a larger number of participants and more time to test. However, our study indicates that the participants were able to gain an understanding of the topic using SimpliCT. We are therefore able to conclude that SimpliCT has the possibility of being effective as a digital tutor.

Furthermore, our study has revealed areas that affect the usability of digital tutoring chatbots, e.g., structure and length of responses, usage of examples and natural language.

¹<https://dev.botframework.com/>

²<https://chatterbot.readthedocs.io/en/stable/>

³<https://platform.openai.com/docs/overview>

We can conclude that a digital tutor should be trained to the specific topic in hand. Furthermore, an acceptable user experience will ensure that students will use the chatbot and find the interaction more exciting.

SimpliCT: A Digital Tutor Designed to Aid in the Education of Computational Thinking

Jacob G. Jensen* and Jonas J. Svare*

*Aalborg University, Department of Computer Science, Denmark

Abstract—Computational Thinking (CT) is a concept that involves solving problems, designing systems and understanding human behaviour, by drawing on fundamental Computer Science (CS) concepts. We believe it has the potential to empower individuals in their ability to solve everyday problems and improve the academic competencies.

In this project, we seek to find out how a digital tutor can assist in tutoring computational thinking. We develop a chatbot called SimpliCT, using a GPT model provided by OpenAI. We have fine-tuned the model with 68 training examples.

We conduct a comparative study to determine the user experience and effectiveness of SimpliCT. We found the user experience of SimpliCT to be acceptable with possibility for improvement. We are not able to determine the effectiveness, but our study indicates that new learners are able to obtain knowledge about CT.

I. INTRODUCTION

COMPUTATIONAL THINKING is a concept that has gained traction within the educational world in the last twenty years. CT involves solving problems, designing systems and understanding human behaviour, by drawing on fundamental CS concepts [22]. CT is about the deconstruction of a seemingly difficult problem into simpler problems we know how to solve, relying on decomposition and abstraction of the problem [23].

One of the pioneering researchers of CT, Jeanette Wing, states the characteristics of CT which can be categorised into four components; decomposition, pattern recognition, abstraction, and algorithm design. [23] Decomposition is about breaking a problem into smaller, more manageable parts. This allows for focusing on one part of the problem at a time, as well as identifying tasks that can be delegated within a team.

In pattern recognition, we may discover patterns and structures of the decomposed parts of the problem. These patterns are what form the foundation of the algorithms that we want to create.

Abstraction is about disregarding irrelevant details, only focusing on details that are relevant for solving the problem. According to Grover and Pea (2013), abstraction is the keystone to CT, and is what sets it apart from other ways of thinking [7].

When thinking of algorithms, one might connect it to the highly specified algorithms, written in code, in CS. However, in the context of CT, algorithms are generalized to being replicable lists of step-by-step instructions that solve the problem. The algorithm design component is about creating these algorithms, using the discoveries from the three previous

components.

According to Wing, CT should be considered a formative skill on par with reading, writing and arithmetics [22]. While CT is often thought of as being strictly applicable in CS, this is not case. CT helps us process information and tasks in a systematic and efficient manner [15]. The usecases of CT span broadly, from almost every area of study (particularly in the STEM field), to the daily lives of the general population [14], [19], [22]. This places a significant importance of learning the concepts of CT. We believe it not only has the potential to empower individuals in their ability to solve everyday problems, but also to improve the academic competencies in the educational system, as CT competencies allow individuals to handle complex problems.

The recent surge in popularity of conversational AI agents, or chatbots, has been met with both excitement and concern. One area in which this concept has been particularly discussed, is education. Conversational AI and chatbot technologies have the potential to capture and prioritise the needs of the individual student, as they interact directly with individuals, allowing for adjustment to their needs [24].

Okonkwo & Ade-Ibijola (2021) has made a systematic review of chatbots used in education to determine the variety of ways these chatbots benefit the educational system [18]. They found that chatbots can assist educators in providing students with information tailored to specific courses, as well as providing students easy access to course content. Furthermore, they found that chatbots promote quick access to required information, which helps with time optimization, as well as maximising the learning abilities and achievements of the student.

Okonkwo & Ade-Ibijola also discovered how chatbots keep students motivated and engaged. Conversational agents allow students to acquire knowledge in an exciting manner, resulting in increased student engagement. This is strengthened by the fact that chatbots can provide quick support during individual learning, provide immediate answers, and adapt to learners' actions.

Due to the educational potential of chatbots and importance of learning CT, this project will focus on the development of a chatbot prototype to aid students by acting as a digital tutor of CT. As such, we present the following problem statement:

How can a digital tutor assist in tutoring computational thinking?

We consider our main contribution to be our findings from answering our problem statement, as they provide an understanding of what features may be important to a chatbot, in order to be educationally beneficial.

The rest of this project is structured as follows: In section II, we seek to gain knowledge of both chatbots and CT in an educational context. Section III presents the prototype we have developed using the knowledge we gained. Our research questions are presented in section IV, which we seek to answer through the means described in section V. In section VI, we present our results and discoveries from this project, while sections VII, VIII, and X provides a discussion of our findings, and the limitations and possible future work of the study, respectively.

II. RELATED WORK

A. Educational Chatbots

Several studies have been conducted to research and develop chatbots used in education. According to Vanichvasin (2021), chatbots can be useful in an educational setting by supporting and facilitating the teaching process. Chatbots can act as virtual assistants or tutors, allowing students to ask it simple or repetitive questions, freeing up the teachers to go in depth with more complex matters. The usecases for chatbots in educational settings rely on them acting as digital learning tools to ask questions, give answers, retrieve information, provide useful information, establish knowledge and propose possible solutions [20].

Weber (2005) introduced a rule-based chatbot called VPbot, which can be used in specific contexts [21]. Weber points to its usage in the Harvard Medical School, where it was used to simulate patients, for medical students to interview through a web-based interface [21]. In their investigation of the effects of web-based teaching, Kerfoot et al. (2006) found that medical students who used VPbot showed a significant increase in both test scores and learning efficiency [12].

Other studies show benefits in various areas for students utilizing chatbots in educational contexts. Deng et al. (2023) concludes that chatbot technologies are able to improve learning achievement, as chatbots can notice gaps in the learners knowledge and accordingly adjust responses in order to create meaningful conversations [3]. Deng et al. also found conversational behaviors among students becoming more transactive, i.e., with more explicit arguments, when students use a chatbot. Furthermore, they found that chatbots could improve retention, by randomly generating multiple-choice questions for declarative knowledge testing and open questions for procedural knowledge testing [3].

Labadze et al. (2023) notes several educational areas where chatbots are advantageous [13]. Students have the ability to receive study and homework assistance from chatbots acting as tutors, providing guidance with course material or homework problems, or quizzing the student on subjects

within the course. Labadze et al. states that chatbots are capable of adapting their teaching approach such that they are able to provide individualized guidance and feedback for a more flexible learning experience. This can improve motivation in students, especially if they already experience learning problems.

The development of a students academic skills is also something that can benefit from chatbot usage. Chatbots are able to suggest corrections to syntax and grammar, improving writing skills. By providing discussion structures and real-time feedback, chatbots are also able to help with group discussions and debates [13].

While the benefits of including chatbots in education are plentiful, some challenges have also been identified. Labadze et al. highlights how a students problem-solving and critical thinking skills could be negatively impacted by an overly excessive reliance on chatbots. This problem stems from "improper" use of the chatbot, e.g., by relying on the chatbot to provide information or solutions without taking the time to understand the topic at hand [13].

Another challenge of using chatbots in education is the difficulty of verifying the output given by a chatbot (Kasneji et al. (2023) [11], Elkins & Chun (2020) [4], Gao et al. (2022) [6]). The human-like language used by many chatbots makes the information they provide seem convincingly real. However, due to the massive amount of data used to train these chatbots (the GPT-3 model was trained with upwards of 45TB of data [8]), some of it is bound to be incorrect. This can potentially result in students believing false or misleading information, without examining its authenticity [11].

With the ability to get chatbots to generate large amounts of text, the concern of cheating and plagiarism in the educational system has been rising along with the availability of chatbots (Cotton et al. (2024) [2], Gao et al. (2022) [6], Nassim (2021) [16]). Using a chatbot, students are able to quickly and effortlessly generate essays, and solve other assignments. This relates to the two other challenges described above, in the sense that the students are not putting in the effort to learn from the tasks they are given, and they are not able to validate the text they do hand in; neither in its correctness, nor in the plagiarism in the text.

B. Computational Thinking in Education

Hammond et al. (2019) made a study focusing on ways to adapt the concepts of CT to the context in which it is used. The study was a social study, conducted on third grade students during a scaffolded geocache geography lesson, to demonstrate the inclusion of CT [9]. Hammond et al. found that students showed signs of CT when solving problems, which can be correlated to the concepts described by Wing (2006, 2008) [22], [23]:

- **Decomposition:** When the students split up in pairs to focus only on latitude or longitude.
- **Pattern Recognition:** Students would recognize patterns when certain actions were performed, e.g. when moving north, the latitude would change.

- Abstraction: The use of latitudinal and longitudinal coordinates.
- Algorithm Design: The process of navigating to their targets can be seen as following an algorithm the students have designed themselves.

Hammond et al. showed that CT appears subconsciously and is used naturally when solving problems, even as early as the third grade. Furthermore, it was suggested how CT can be adapted into the context of social studies, while still maintaining its core concepts, indicating that CT can be incorporated into different areas of education.

The notion of broadening CT to fit other educational fields was strengthened by Li et al. (2020) in their study, which described the need to re-conceptualize CT, particularly in the field of STEM education [14].

While Li et al. goes into detail about different approaches of developing a CT mindset, their education-oriented approaches are highly relevant to us. These approaches describe the relevance of facilitating the education of CT, due to the vast amount of perceptions of it. The first approach is discipline-based, going through each concept step by step. This approach puts emphasis on efficiency and elegance when designing a solution to a problem. Li et al. do note that this approach advocates more to the area of CS, as it requires solutions to be laid out in computational steps, which may require programming experience.

The second approach focuses on expanding and adapting CT beyond CS, similarly to how Hammond et al. (2019) showed that CT could be adapted to the specific context in which it is used [9]. This approach can also be seen in the 2018 report released by the U.S. White House; *Charting a Course For Success: America's Strategy for STEM Education*. The report broadens the general definition of CT into: "evaluate information, break down a problem, and develop a solution through the appropriate use of data and logic" [19]. It proposes using this broader definition of CT as a foundation in all areas of the U.S. education system.

These approaches can be useful for us to determine our contribution. If we can create a solution which incorporates CT, and applies it to the relevant fields, while still retaining a clear, discipline-based manner, we believe it will improve the academic competencies of the current education system.

III. PROTOTYPE

The findings in section II suggests a chatbot to be a viable tool for tutoring students learning CT. In this section, we introduce our prototype, SimpliCT, a chatbot designed to act as a tutor aiding help with teaching CT. The objective of SimpliCT is to allow users to be able to ask it questions regarding CT, and get relevant responses in return. SimpliCT is a generative chatbot, meaning that responses are generated by AI and are therefore not predefined based on certain input. To allow for SimpliCT to be easily available, interaction with it is available via a webpage, developed in the React library

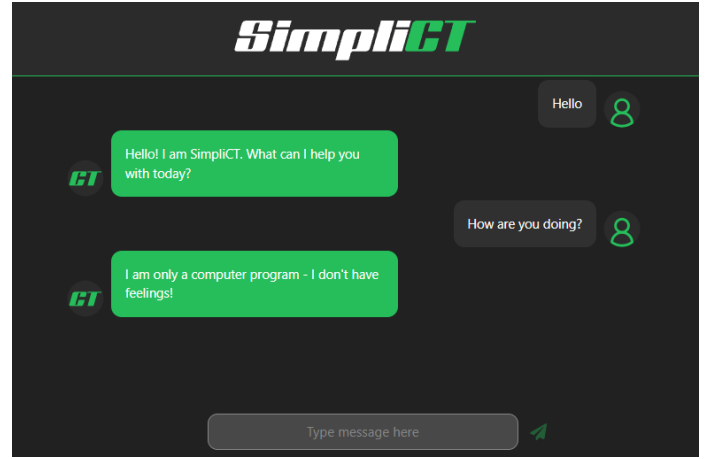


Fig. 1: The user interface of SimpliCT

for JavaScript. The Chatscope¹ UI Toolkit, has been used to quickly implement a simple UI.

To ensure its simplicity, SimpliCT consists of a single page, seen in Figure 1, designed as a regular chat interface, which most users should be familiar with. The interface of SimpliCT is inspired by other conversational platforms, such as *Messenger* and *ChatGPT*.

With the use of AI being popular, and numerous models being readily available, we have chosen to use an existing generative pre-trained transformer (GPT) model provided by OpenAI². Using the GPT model should result in more accurate responses and satisfying user experience, as OpenAI have spent numerous years developing this model. SimpliCT specifically uses the `gpt-3.5-turbo-0125` model which, while not being the latest and most knowledgeable model, is the latest model allowing for fine-tuning, further described in subsection III-A.

When a user sends a message to SimpliCT, the message is added to a list of messages, representing the entire conversation. The list of messages is then sent to the API of the model via a POST request. The answer is then awaited and added to the list of messages and the UI is updated accordingly. Sending the entire conversation allows for the model to provide more accurate answers, as it provides context of the topics that have been included in the conversation.

A. Fine-tuning

Fine-tuning is used to specialise SimpliCT to adhere to the material provided in the CT course, and ensure more relevant responses. The model has been fine-tuned by creating a dataset containing a list of conversational training examples. A conversational example consists of a suggested user prompt and the ideal response to the prompt. Furthermore, a conversational example contains a system message, which specifies the "personality" of the model. The system message allows the bot (referred to as an "assistant" by OpenAI) to act more

¹<https://chatscope.io/>

²<https://openai.com/product>

realistically, as its behaviour is adaptive to the context of the conversation.

This is an example of one of the conversations used to fine-tune the model:

```
{"messages": [{"role": "system",
"content": "You are SimpliCT, a
chatbot designed to act as a helpful
tutor whose purpose is to help with
teaching computational thinking."},
{"role": "user", "content": "What is
computational thinking?"}, {"role":
"assistant", "content": "Computational
thinking is the process of going
from a complex problem and ending
up at a clear step-by-step solution
that solves parts or the problem
as a whole. Computational Thinking
can be divided into four parts:
Decomposition, abstraction, pattern
recognition, and algorithm design."}]}
```

In total, our fine-tuning dataset consists of 68 training examples similar to this, with different user prompts, and relevant responses coming from the assistant. The complete dataset can be found in the attached appendix file.

IV. RESEARCH QUESTIONS

To answer the problem statement, presented in section I, we have identified two research questions. These will be answered using the prototype presented in section III. To ensure the capabilities of SimpliCT functioning as a digital tutor, we consider User Experience (UX) and effectiveness to be meaningful and relevant aspects.

UX is deemed as an important aspect, since it will affect how inclined students are to using the chatbot. Therefore, RQ_1 addresses this:

RQ_1 : What is the UX of SimpliCT?

An educational chatbot should be able to provide information and allow for students to get an understanding of CT. Therefore, we want to ensure that SimpliCT has an effect on new learners of CT, which has led to RQ_2 :

RQ_2 : Is SimpliCT effective in tutoring new learners of CT?

V. THE STUDY

In this section, we describe how we conducted our study. The purpose of the study is threefold. First, we want to examine how users perceive SimpliCT. Second, we want to find out if SimpliCT actually helps new users understand CT. Third, we want to examine what SimpliCT does right and what could be improved.

In this study we presented two participant groups, one with prior knowledge of CT and one without, to SimpliCT,

and asked them to complete a series of tasks. Then we presented them with a UX questionnaire, in order to address RQ_1 . The group with prior knowledge of CT was interviewed, while the group without experience was presented to an exam set in order to address RQ_2 .

All participants signed a consent form prior to the study, and will remain anonymous.

A. Recruitment and Participant Demographics

The study was conducted on 14 participants, divided into two groups, based on their experience with CT. The reason for having two groups, is that we want to obtain knowledge about how people without prior knowledge of CT obtained an understanding of CT, by using our prototype. Participants were recruited through convenience sampling [5], as we specifically wanted two separate groups with and without experience with CT.

Group A consisted of 7 participants without experience within CT. There were no criteria regarding occupations, educational levels, experience with chatbots, etc. This was a deliberate choice as prior studies noted how the concept of CT should be beneficial to all members of society. The only requirement were that participants had no prior knowledge of CT.

Group B consisted of 7 participants. The common feature of group B, was that the participants had completed the CT course at the Computer Science Department at Aalborg University. As all participants of group B had previously attended the CT course, we assumed they knew how the material should be properly presented in order for new students to learn.

B. Tasks

Both Groups were asked to complete the following tasks to ensure that participants became familiar with SimpliCT:

- 1) Use SimpliCT to learn what CT is.
- 2) Identify the four basic processes of CT.
- 3) Use SimpliCT to learn what Decomposition is.
- 4) Use SimpliCT to learn what Abstraction is.
- 5) Use SimpliCT to learn what Pattern Recognition is.
- 6) Use SimpliCT to learn what Algorithm Design is.
- 7) Solve an example problem using CT. First, Explain how you come up with your solution using the concepts of Computational Thinking. Second, use SimpliCT to organize the books and compare your answers. The example question is as follows:

"Imagine that you have an empty bookshelf, and a large pile of unsorted books. How would you organize your books and insert them into your bookshelf?"

C. Questionnaire

With all tasks completed, we used the online questionnaire tool, AttrakDiff³, to evaluate the user experience of each participant. We used AttrakDiff as it is a renowned standardised

³<http://www.attrakdiff.de>

Test scores	7-point grade	ECTS grade
90-100	12	A
77-89	10	B
57-76	7	C
42-56	4	D
33-41	02	E
0-32	00	Fx

TABLE I: Grading scale of the group A test, and its conversion to Danish 7-point scale and ECTS scale. The '-3' grade was removed as it would not be relevant to the grading of this test.

test to evaluate user experience. The AttrakDiff questionnaire allowed for us to judge the pragmatic and hedonic qualities of SimpliCT [10], while distinguishing between the groups. This allowed for us to compare the findings between group A and B.

The questionnaire measures four dimensions:

- **Pragmatic Quality (PQ):** The usability of the product.
- **Hedonic-Stimulation Quality (HQ-S):** Stimulation generated by the system.
- **Hedonic-Identification Quality (HQ-I):** The identification of the user with the system.
- **Overall Attractiveness (ATT):** The overall attractiveness of the system, based on pragmatic and hedonic qualities.

We made use of the 28 item version of the AttrakDiff questionnaire.

D. Group A Test

Each participant in group A was given a written test, included in Appendix B, where they were asked to explain CT terminology and solve different CT-related tasks. The purpose of the test was to study if participants understood CT, when it had been explained by SimpliCT.

Each question in the test was worth a certain amount of points, totalling 100 points. The score for each question was evaluated by how well the participant showed an understanding of the concept they were asked to explain, and how well they were able to put it to use. As such, the score for a question could be anywhere between 0 and the total score for that question. How points were converted into grades, can be seen on on Table I.

We graded the tests on the Danish 7-point grading scale, in order for us to compare the average grade with the average exam grade of the previously held CT course.

E. Group B Interview

An interview was only conducted with the participants from group B, as they already had experience with learning CT and we therefore could consider them to be able to estimate what would be helpful for understanding and learning CT. We did not interview participants from group A, as we considered them unable to estimate how well CT was explained.

The purpose was therefore to identify what the participants of group B thought worked and what could be improved in order for SimpliCT to function as a virtual tutor.

The interview was set up in a semi-structured manner, which allowed us to stay on the topic of the questions, while

exploring the ideas that may arise from them. The questions that the participants were asked, were as follows:

- 1) Did you encounter any errors?
- 2) If any, what parts did you like about SimpliCT?
- 3) If any, what parts did you dislike about SimpliCT?
- 4) Do you think it would be advantageous for people attending the CT course to have SimpliCT available?

VI. RESULTS

A. RQ1: What is the UX of SimpliCT?

As previously mentioned, both groups were presented with the AttrakDiff questionnaire. The results of the questionnaire can be seen in Figure 2. Results from group A and B were very similar, with both groups giving an average PQ rating of 1.47. Group B gave an average HQ-I rating of 0.55, where group A gave an average of 0.41. The biggest difference between the groups were in the rating of HQ-S, where group B gave an average rating of 0.20, while group A gave a rating of -0.49. In the ATT dimension, group A gave an average rating of 1.04, while group B gave it 0.94.

Separate result values for each individual question can be seen in Appendix A.

B. RQ2: Is SimpliCT effective in tutoring new learners of CT?

Each participant in group A was given a test to determine how well they had understood CT. Each submission was graded, and converted to the Danish 7-point scale. The grades of each participant can be seen on Table II.

We see three participants achieving a grade of 7, with the remaining four achieving a grade of 4.

The average point score was 58.1, while the average grade was 5.3.

C. Group B Interview

During the interview with group B, we gained insight into the views and opinions of SimpliCT, from the viewpoints

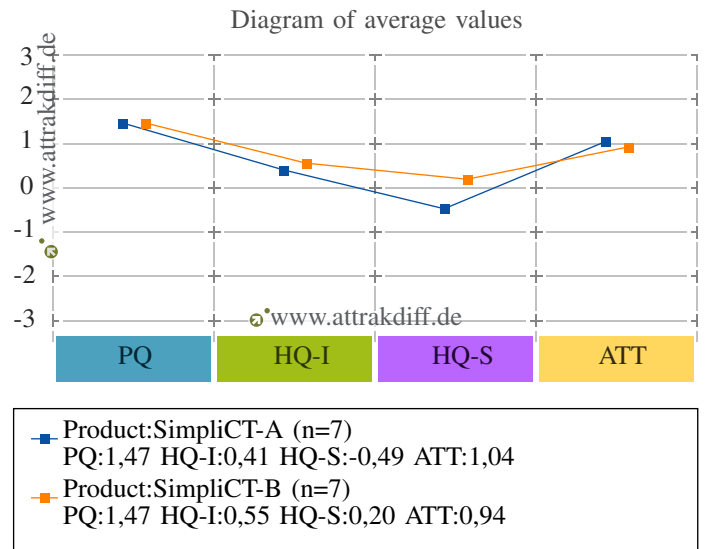


Fig. 2: The results of the AttrakDiff questionnaire.

Participant	Total / 100	Grade	ECTS grade
1	51	4	D
2	71	7	C
3	73	7	C
4	47	4	D
5	47	4	D
6	47	4	D
7	71	7	C
Average	58.1	5.3	C

TABLE II: The test scores and grades for each participant in Group A.

Category	#	Answer
No	4	
Yes	3	Minor issue - missing "."
		When asked about the four main concepts, it only listed 3: decomposition, pattern recognition and algorithm design. Abstraction was introduced as an underlying point to Pattern recognition. Kept with the same definition (3 concepts, Abstraction as sub-concept) in the book collection example.
		Asked the first question in Danish, but the rest in English. This may have resulted in errors in the layout of the rest of the conversation.

TABLE III: The categorizations for the answers given to the interview question: "Did you encounter any errors?". The answers to the "No" category has been omitted as no further explanations were needed for the answers.

of people experienced with CT. Since the questions were open-ended, answers were phrased differently. Therefore, we have categorised and grouped similar answers. Answers were considered to be similar if they focused on the same topic or element.

1) *Errors*: When asked about possible errors that the participants encountered, four participants stated that they found no errors, while three answered "yes". One participant noted how SimpliCT would give a wrong definition of CT by only listing three main concepts; Decomposition, Pattern Recognition and Algorithm Design, with Abstraction being mentioned as an underlying point to Pattern Recognition.

Another participant asked his first question in Danish before switching to English, which may have caused errors in the layout.

The answers to this question are shown in Table III.

2) *Positives*: Table IV lists the categories of answers given by the participants, as well as each individual answer, when asked about what they liked about SimpliCT. 9 different categories were identified, with the "Responses" category appearing the most, followed by "Design", "Examples", and "Response structure".

3) *Negatives*: Table V shows how each answer, given by the participants when asked what they did not like about SimpliCT, was categorized. Most participants found the length of the responses to be problematic, either by being too long and as a result becoming overwhelming, or by being inconsistent in the length of its responses.

We noticed that the participant group seemed to be split between liking the design or not, with three participants saying they liked the design, as shown in Table IV, and three saying

Category	#	Answer
Design	3	Simple interface
		Looks familiar to other chatbots
		Color choice was good
Responses	4	Comprehensive answers
		In-depth answers
		Great explanations
		Likes that it automatically provides answers
Natural language	2	Feels natural to talk to - like any other chatbot
		Nice language - natural formulation (not too professional / technical)
Examples	3	Nice examples - Helps with understanding
		Great examples - good for new people to learn
		Gives real-world examples, really helps to give an understanding of the topic
Ease of use	2	Easy to figure out what to do
		Simple usage - not too clustered which avoids confusion
Useful outside CS	1	Gives explanations in a way that people with no CS background should be able to understand
Trained for CT	2	Seems trustworthy, as it is trained for CT specifically
		Likes that the bot was specifically trained for CT
Response structure	3	Nicely structured
		Well structured
		Nice that answers are structured in points
Expected functionality	2	Functions as expected
		Asked to explain it like I'm 8, functions as expected

TABLE IV: The categorizations for the answers given to the interview question: "If any, what parts did you like about SimpliCT?".

they did not like it.

"Confusing layout", "too little detail in answers", and "design issues" were also mentioned by several participants. Only one participant mentioned "incorrect information", "No examples" and "too focused on problem solving".

4) *Advantageous for CT course*: Table VI shows the responses from the participants when asked if they thought having SimpliCT available would be advantageous for students in the CT course, and their categorizations. Five participants answered "yes", albeit two of them suggested that modifications should be made before they could fully agree. One of which, expressed that they would have to be ensured that the information they receive is correct, before it can be used in education.

Two participants did not think that SimpliCT would be advantageous in the CT course. One participant stated that it would be more useful in a context where it was needed to solve a problem. The other participant said that it did not facilitate learning, as it was too quick to give out answers instead of leading the user to figure out the problem themselves.

VII. DISCUSSION

In this section, we discuss and speculate on the results of our study, and provide incentive for future research of digital tutors in the CT field.

A. The User Experience of SimpliCT

In our study we found a slight difference in the user experience scores of group A and group B.

Comparing the results from group A and B, we can see that both groups found the efficiency and ease of use of SimpliCT to be at an acceptable level. This can be seen as both groups evaluated the PQ to be 1.47. In subsection VI-C, two participants from group B said that they found SimpliCT to be simple to use. One stated: *"Easy to figure out what to do"*, the other stated: *"Simple usage - not too clustered, avoids confusion"*. However, a participant from group B said that they did not always notice when SimpliCT had given a response. The participant even suggested using some kind of animation to notify the users, when an answer had been provided.

We also found that group B felt a slight stronger identity connection with SimpliCT, than group A. Group A evaluated the HQ-I to be 0.41, while group B evaluated it to be 0.55. The difference between the groups could be caused by the language used by SimpliCT. Participants from group A, might have found the language to be a bit technical. In the interview, presented in subsection VI-C, two participants (from group B) stated that they found the language to be natural, with one specifically stating that the language was not too technical.

The most significant difference in the ratings is seen with the HQ-S, with group A giving it a rating of -0.49 while group B gave a rating of 0.20. This indicates that group B found it more exciting to use the chatbot. One reason for group A giving such a low rating might be due to the ambiguity of some of the questions in the questionnaire. Particularly, with the "undemanding - challenging" question, it is not totally clear that AttrakDiff views "undemanding" as a negative trait. On the contrary, someone without experience with a system might look positively on it being undemanding. When looking at the individual ratings for each question, we see that group A rated the system as being more undemanding than group B. We believe that stems from the fact that group A might have seen it as a positive trait.

We found no significant difference in ATT ratings between group A and group B. The results indicate that both groups found SimpliCT visually pleasing. However, in the interview, presented in subsection VI-C, the participants of group B gave somewhat conflicting opinions of the design of SimpliCT. Three participants stated that they liked the design, while three others did not like it, stating that it was boring, contained distracting colors and lacked a way for identifying when new responses were generated.

Overall, the results show that the UX of SimpliCT is acceptable, but could be improved. Through the interview with group B, we have identified these possible improvements:

- Giving shorter and more clear responses.

Category	#	Answer
Response length	4	Long answers
		Sometimes answers are very long
		Long answers were overwhelming
		first answer very short, compared to other answers (might not be a problem with using the bot in a project or other context)
Response structure	2	Didn't write descriptions in bullet points, but wrote walls of texts which was confusing
		The layout, when using SimpliCT to solve the example problem, was weird
Detail in answers	2	When asked about an explanation of the 4 constructs, it gave examples instead of actually explaining them
		Goes into too little detail in the answers it gives
Incorrect information	1	In the example problem, it mentions a step "Problem Solving", which was not mentioned previously, either by the bot or in the course
Design	3	Boring design
		Design (colors) are a bit distracting
		Could make use of an animation, when an answer has been generated - didn't always notice when a new answers were given
No examples	1	Did not give examples without being asked to
Too focused on problem solving	1	The descriptions given beforehand, did not help/created confusion when solving the example problem (without using SimpliCT) - Too oriented towards solving concrete problems instead of explaining the actual theory behind CT

TABLE V: The categorizations for the answers given to the interview question: *"If any, what parts did you dislike about SimpliCT?"*.

Category	#	Answer
Yes	3	Yes - gives examples, inspiration, and context
		Yes, the topic is new - so it's a great tool for learning terminology in a simple way. Can be useful in projects
		100 Percent, I already use a chatbot to verify own notes in relation to the course material
Yes - with modifications	2	Yes, if I could be assured that the data is correct. Easier than reading book or own notes. Great examples, nice solution help
		Yes - Responses could be more concrete though
No	2	May not help, as the topic was brand new. Might be more useful when you have to use it in a project. The bot is great for helping using CT in a context where you have to solve a problem
		Does not facilitate learning. It gives straight answers, instead of "provoking" the student to learn

TABLE VI: The categorizations and answers provided by the participants when asked if they believed SimpliCT would be advantageous in the CT course.

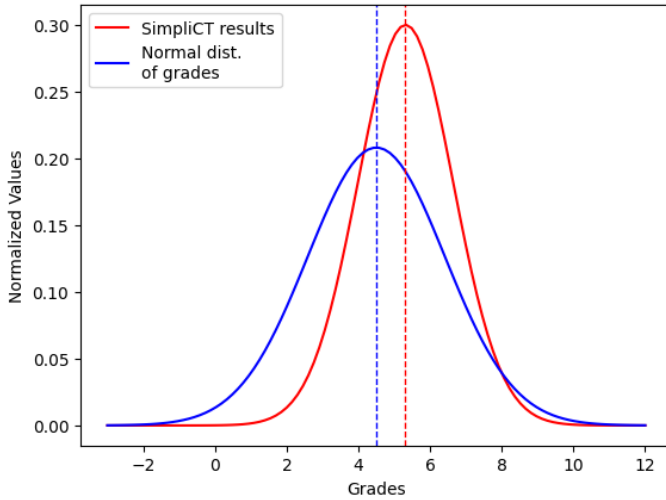


Fig. 3: A visualization of the results from the test performed by group A.

- Split responses into bullet points, avoiding walls of text.
- Create a more exciting design.
- Make use of animations.

B. The Effectiveness of SimpliCT on New Learners

In our study we found the average grade of the participants without prior knowledge in CT to be 5.3.

Unfortunately, the CT course was a pass/fail course, making it impossible for us to directly compare our results with the course results.

Instead, we assume that the course follows the normal distribution of the Danish 7-point grading scale. Our results were fitted into a Logit-Normal distribution [17] to let us compare our results with the normal distribution. The distributions can be seen on Figure 3.

By comparing the two distributions, we see a wider spread in the normal distribution. The narrow spread in our results, is caused by the limited amount of data. Furthermore, we see that the distribution of our results peak at 5.3, while the normal distribution of grades peak at 4.5.

Our results indicate that SimpliCT is actually able to provide users with an understanding of CT and, by extension, give competencies in applying it in different situations. We can relate our results to the findings of Kerfoot et al. (2006), who found that a chatbot significantly increased test scores [12].

C. Usefulness of SimpliCT as a Digital Tutor

In section VI-C4 we found that a majority of participants from group B stated that SimpliCT, with slight modifications, has capabilities of functioning as a digital tutor in the CT course at AAU. Most participants liked the "Responses", "Natural language", "Examples", and "Response structure" categories, making up 10 out of 22 answers. According to prior studies, having well-structured and clear responses could help students gain an understanding of the subject at hand [3], [13].

A concern, which was only mentioned by one participant, is that SimpliCT does not facilitate a deeper understanding of the

subject. While most participants did like the responses they received, they were never guided to figure out a solution to a problem for themselves. This correlates with the challenges of using chatbots in education that was identified by Labadze et al. (2023) [13]. SimpliCT could directly feed into what Labadze et al. describes as "improper usage" of chatbots, by providing solutions to a problem straight away without ensuring that the user gains an understanding of the subject.

VIII. LIMITATIONS

A. Number of Participants

The limited amount of participants used for the study made it impossible for us to give a definitive statement of the functionality of SimpliCT as a digital tutor.

B. Learning Retention

As time was a constraint, participants in group A only had access to SimpliCT for a short period of time. Furthermore, they were given the test shortly after they had access to the SimpliCT. As participants only had limited time to interact with SimpliCT and the short period of time between receiving information and test, we were not able to ensure learning retention [1]. We were not able to determine if the information provided by SimpliCT is stored in the participants short- or long term memory.

C. Training Data

OpenAI describes clear improvements of fine-tuning with 50-100 examples⁴. The model was fine-tuned using 68 training examples, all containing information from the course. With more training examples, SimpliCT could have been more specified and potentially provide better responses.

IX. CONCLUSION

Our findings and discussions help answer the initial problem statement: "*How can a digital tutor assist in tutoring computational thinking?*".

To answer this, we have developed SimpliCT, a simple generative chatbot designed to tutor people in CT. We have fine-tuned the gpt-3.5-turbo-0125 model, using a dataset containing 68 training examples.

The SimpliCT prototype was used to test how digital tutors can assist in tutoring computational thinking. This led to the creation of two research questions which were answered by conducting a comparative study on two groups, one with and one without CT experience.

RQ₁ : What is the UX of SimpliCT?

This question was answered by presenting an AttrakDiff questionnaire to both groups, after they had used the prototype. The results from the questionnaire lets us conclude that the UX of SimpliCT sits at an acceptable level.

RQ₂ : Is SimpliCT effective in tutoring new learners of CT?

⁴<https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset>

This question was answered by testing the group without CT experience after they had used SimpliCT. The answers were graded, fitted into a Logit-Normal distribution and compared with the normal distribution of the Danish 7-point grading scale. Given the nature of our study, we are not able to prove that students are able to learn CT from SimpliCT, as this would require a larger number of participants and more time to test. However, our study indicates that the participants were able to gain an understanding of the topic using SimpliCT. We are therefore able to conclude that SimpliCT has the possibility of being effective as a digital tutor.

We can conclude that a digital tutor should be trained to the specific topic in hand. This can be done by training the model on data strongly related to the topic. Having an acceptable user experience can ensure that students will actually use the chatbot and find the interaction more exciting.

Our study has revealed areas that affect the usability of digital chatbots, e.g., structure and length of responses, usage of examples and natural language.

Our findings and discussions contribute to improving the development of digital tutoring tools, as it gives an understanding of what affects learning, while using a digital tool such as a chatbot.

X. FUTURE WORK

Future studies should be performed with more participants in order to get more precise data. This should give better opportunities in detecting meaningful differences between the two groups. Furthermore, future research of the effect of chatbots as digital tutors should ideally be conducted over an entire course. One possibility of such, could be to run a longer-term comparison study of students in a CT course. Both groups would follow the same course, but one group gets access to the chatbot instead of a human tutor. This would make it possible to compare the final exam scores, in order to prove whether the chatbot is effective in tutoring new learners or not.

With the group B interview revealing some issues with the responses given by SimpliCT, we can assume that the data used to train the model was not sufficient. We found slight variations in the descriptions of CT that SimpliCT would give. The provided training data would describe the four main components of CT as: *Decomposition*, *Abstraction*, *Pattern Recognition* and *Algorithm Design*, in that order. However, SimpliCT would often mix the order of this description, creating confusion with the participants.

Another issue that we found was that SimpliCT provided solutions right away instead of guiding participants to solve problems themselves. Future development should restrict SimpliCT from providing solutions directly. Instead, SimpliCT should facilitate follow-up questions, guiding the user to come their own solution.

However, the results from the interview indicate a number of

things related to the training of our model, that needs to be taken into consideration for future development:

- More training examples may need to be provided to ensure that the descriptions provided by SimpliCT are consistently structured.
- The training data needs to be reconfigured to make sure that SimpliCT is more inclined to facilitate learning instead of giving up solutions immediately.

ACKNOWLEDGMENT

The authors would like to thank the following:

- Dimitrios Raptis, for his continued supervision and guidance throughout this project.
- Martin Lindrup, for providing us with material about CT, allowing us to train our model to adhere to the CT course.

REFERENCES

- [1] Kathleen Davey. Learning retention. <https://www.docebo.com/glossary/learning-retention/>. Accessed: 05-06-2024.
- [2] Peter A. Cotton Debby R. E. Cotton and J. Reuben Shipway. Chatting and cheating: Ensuring academic integrity in the era of chatgpt. *Innovations in Education and Teaching International*, 61(2):228–239, 2024.
- [3] Xinjie Deng and Zhonggen Yu. A meta-analysis and systematic review of the effect of chatbot technology use in sustainable education. *Sustainability*, 15(4), 2023.
- [4] Katherine Elkins and Jon Chun. Can gpt-3 pass a writer’s turing test? *Journal of Cultural Analytics*, 2020.
- [5] Alison Galloway. Non-probability sampling. In Kimberly Kempf-Leonard, editor, *Encyclopedia of Social Measurement*, pages 859–864. Elsevier, New York, 2005.
- [6] Catherine A. Gao, Frederick M. Howard, Nikolay S. Markov, Emma C. Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T. Pearson. Comparing scientific abstracts generated by chatgpt to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. *bioRxiv*, 2022.
- [7] Shuchi Grover and Roy Pea. Computational thinking in k–12: A review of the state of the field. *Educational Researcher*, 42(1):38–43, 2013.
- [8] Sue Halpern. What we still don’t know about how a.i. <https://www.newyorker.com/news/daily-comment/what-we-still-dont-know-about-how-ai-is-trained>. Accessed: 06-06-2024.
- [9] Thomas C. Hammond, Julia Oltman, and Meghan M. Manfra. Geo-computational thinking in the third grade: Making computational thinking truly “for everyone, everywhere”, 2019.
- [10] Julien Hennig. How to use attrakdiff?
- [11] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutytiok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274, 2023.
- [12] B. Price Kerfoot, Harley Baker, Thomas L. Jackson, William C. Hulbert, Daniel D. Federman, Robert D. Oates, and William C. DeWolf. A multi-institutional randomized controlled trial of adjuvant web-based teaching to medical students. *Academic Medicine*, 81(3), 2006.
- [13] Lasha Labadze, Maya Grigolia, and Lela Machaidze. Role of ai chatbots in education: systematic literature review. *International Journal of Educational Technology in Higher Education*, 20(1):56, Oct 2023.
- [14] Yeping Li, Alan H. Schoenfeld, Andrea A. diSessa, Arthur C. Graesser, Lisa C. Benson, Lyn D. English, and Richard A. Duschl. Computational thinking is more about thinking than computing. *Journal for STEM Education Research*, 3(1):1–18, Apr 2020.
- [15] James Lu and George Fletcher. Thinking about computational thinking. *ACM Sigcse Bulletin*, 41:260–264, 03 2009.
- [16] Dehouche Nassim. Plagiarism in the age of massive generative pre-trained transformers (gpt-3). *Ethics in Science and Environmental Politics*, 21:17–23, 2021.
- [17] Arthurs. Noah, Stenhaus. Ben, Karayev. Sergey, and Piech. Chris. Grades are not normal: Improving exam score models using the logit-normal distribution. *Proceedings of Educational Data Mining, Montréal, Canada*, 2019.
- [18] Chinedu Wilfred Okonkwo and Abejide Ade-Ibijola. Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2:100033, 2021.
- [19] Committee on STEM Education. Charting a course for success: America’s strategy for stem education, Dec 2018.
- [20] Patchara Vanichvasin. Chatbot development as a digital learning tool to increase students’ research knowledge. 2021.
- [21] Griffin M. Weber. *Data representation and algorithms for biomedical informatics applications*. PhD thesis, USA, 2005. AAI3174069.
- [22] Jeannette M. Wing. Computational thinking. *Commun. ACM*, 49(3):33–35, mar 2006.
- [23] Jeannette M. Wing. Computational thinking and thinking about computing. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 366:3717–25, 07 2008.
- [24] Shanshan Yang and Kim Stansfield. Ai chatbot for educational service improvement in the post-pandemic era: A case study prototype for supporting digital reading list. In *Proceedings of the 2022 13th International Conference on E-Education, E-Business, E-Management,*

and E-Learning, IC4E ’22, page 24–29, New York, NY, USA, 2022. Association for Computing Machinery.

APPENDIX A
SEPARATE QUESTIONNAIRE VALUES

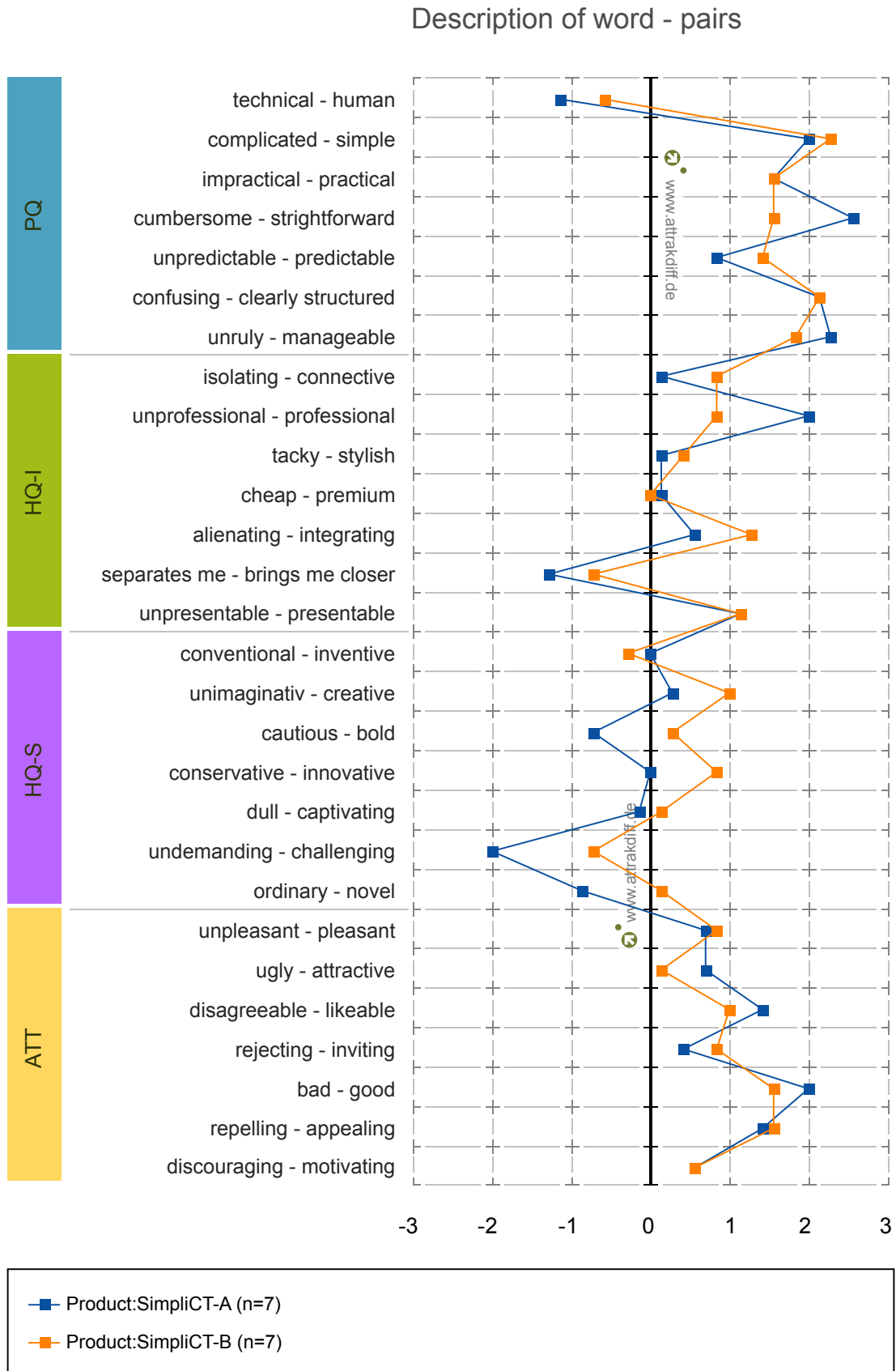


Fig. 4: The results of the separate questions in the AttrakDiff questionnaire.

APPENDIX B
GROUP A SIMPLICT TEST

Question 1: 20 points

In your own words, explain the following Computational Thinking terms:

- (a) Decomposition - 5 points
- (b) Abstraction - 5 points
- (c) Pattern Recognition - 5 points
- (d) Algorithm Design - 5 points

Question 2: 40 points

Consider the following problem:

You have recently purchased a new home. However, the moving company you hired accidentally lost all of your furniture, so now you have the chance to decorate your new home exactly to your liking. The house includes several rooms: a living room, dining room, kitchen, master bedroom, guest bedroom, and a home office.

You decide that decorating an entire house at once is simply too big of a task, so you use your Computational Thinking skills to solve the problem.

Your goal is to furnish each room with appropriate furniture and accessories to make it comfortable and practical.

- (a) How would you decompose the problem? Explain the smaller tasks you end up with after the Decomposition step: - 10 points
- (b) Explain your Abstraction process. What are the less relevant details you choose to ignore? - 10 points
- (c) What patterns do you recognize? - 10 points
- (d) Write down an algorithm that can be used to solve this problem: - 10 points

Question 3: 40 points

Consider the following problem:

After you finished decorating your new house, you still have a sum of insurance money left after paying for your furniture. You decide that you want to spend the left-over money on a vacation. You have multiple destinations in mind and several constraints to consider, such as budget, travel dates, and personal preferences.

After the success of using Computational Thinking to solve your house decoration problem, you choose to use the same method to plan your vacation.

Your goal is to design a comprehensive vacation plan that optimizes your experience while staying within your constraints.

- (a) How would you decompose the problem? Explain the smaller tasks you end up with after the Decomposition step: - 10 points
- (b) Explain your Abstraction process. What are the less relevant details you choose to ignore? - 10 points
- (c) What patterns do you recognize? - 10 points
- (d) Write down an algorithm that can be used to solve this problem: - 10 points