"Oh sorry, I thought you were a chatbot 😳"

IN5480 - Group assignment - Group 5

1. Group description

Our group consists of Bendik Johann Kroken, Chris Kløv Andersen, Inger Helene Howells Engebretsen, and Viljar Tornøe. We are all fourth-year students doing our masters in Informatics: Design, use, and interaction. Bendik, Chris and Inger Helene did their bachelor studies at the University of Oslo, while Viljar did his bachelor's in New Media at the University of Bergen.

2. Area of interest

We would like to work with chatbots. Specifically, we want to investigate the way users interact with chatbots. We want to look at how users choose to formulate their questions when interacting with chatbots versus real people. We consider looking into whether the type of recipient influences the users' vocabulary, sentence structure, and expressions. In order to do this, we want to look at how users interact with the chatbot ToastBot that we made for the student association Toastjærn earlier this semester.

It would also be interesting to look at the expressions the chatbot uses compared to what a human uses, but since ToastBot does not generate its own sentences, this will not be relevant to us in this task. All sentences that ToastBot writes are written by the developers. The chatbot only recognizes certain keywords and replies with the answer that is connected to that specific buzzword.

We are interested in this topic because we all have experiences with either being mistaken for chatbots (through work) or experience using chatbots ourselves. Chatbots are increasingly becoming a larger and more important part of how users interact with companies and this, as Brandzæg and Følstad says, will pose an array of new challenges to HCI (Brandzæg & Følstad, 2017:38-40). Therefore we wish to investigate this concept, and gain insight into the experience of interacting with chatbots, here through language. Another interesting aspect of chatbots and their interaction with users is how the users expect the chatbot to behave. Jenkins et al. (2007), argues that users expect chatbots to both behave and communicate like humans, creating new challenges (Jenkins et al. 2007:83). Drawing upon this we could investigate how this claim carries over to Toastjærns chatbot.

The users we want to include are users who are in the target group for the Toastjærn association. Since Toastjærn is an association affiliated with the Institute of Informatics, it would be interesting to focus on students at IFI. We think it would be interesting to include people who do not necessarily know too much about the student association. If they do know a lot about Toastjærn already, the conversation might not be as natural or organic as it would be if they actually had genuine questions about the association.

In order to make the conversations as organic and natural as possible, we would like to test the chatbot in a natural setting. That will most likely be during lunchtime in the cafeteria at IFI. It is also possible that the chatbot is used during classes or while walking in the hallways, but this might be harder to study. We also do not want to encourage students to use the chatbot during class, even though this might give us an even more accurate example of how users talk to chatbots, especially when in a hurry. We would like to approach students who are eating or socializing in either the hallway or the cafeteria to not disturb their studies.

3. Questions and hypothesis

We would like to investigate this question:

> *"Are users less formal when they know they are chatting with a robot compared to when they think they are chatting with a human?"*

Our hypothesis is that they are. Through our project, we would like to either confirm or disconfirm this. Our hypothesis is therefore this:

> *"Users are less formal when they know they are chatting with a robot compared to when they think they are chatting with a human."*

4. Background

We want to investigate whether and to what extent the language users use change when talking to a robot in comparison to talking to another human being. This is a question that has been addressed by multiple scholars and tech-interested journalists. However, there is not a consensus about whether we should be polite when interacting with artificial intelligence or not. While the journalists Needleman from CallerCallsBack.com and Elgan from FastCompany.com have taken clear stances on what they mean is the right way to interact with artificial intelligences, the scientific community, on the other hand, has not reached a clear stance on the matter (Elgan and Elgan 2018; Gupta, Walker, and Romano 2007; Needleman 2017). In our study, we aim to investigate this phenomenon further and look at how people actually interact with a chatbot in their daily practice.

Both Gupta et al. and Benotti & Blackburn have investigated politeness in human-robot interaction (Benotti and Blackburn 2016; Gupta, Walker, and Romano 2007). These studies were made on the background of people viewing robots and social actors, and thus new issues arose - how polite does one need to be when interacting with robots? Gupta et al. conclude with no clear cut answer to this question, but reports from their studies that there is a cultural component to the subject at hand underlining that politeness with conversational agents varies across both language and the embodiment of the responses of the conversational agent (Gupta, Walker, and Romano 2007).

Politeness is highly contextual (Benotti and Blackburn 2016), and as Luger and Sellen argue, chatbots often lack this contextual information, making interactions with conversational agents seem "patchy" and "off" (Luger and Sellen 2016, 5288). Kocielnik et al. argue that expectations a central tenet in our interactions with conversational agents (Kocielnik, Amershi, and Bennett 2019). Benotti and Blackburn argue that a central part of politeness theory is for the actor (the one chatting) to understand the desires and intentions of the agent, thus prompting a polite response from the actor (Benotti and Blackburn 2016, 301). This is especially interesting in the context of robot-human interaction when this is something that cannot be done, and robots do not have desires/intentions in the way humans do. Relating to our research question, we view politeness as a central part of formality, thus we think it is interesting to use the theories proposed by Benotti and Blackburn.

These authors create a foundation in which we aim to understand our findings. Does a lack of context and immediate responses make a reduction in politeness when interacting with chatbots, or does the opposite happen? Does expecting a robotic reply from a conversational agent triggers a more robotic response from the user?

5. Methods

In order to find out whether the formality of user language differs between interaction with chatbots and interaction with humans, we want to ask students at IFI to chat both with the chatbot “ToastBot” and with a person from the board of Toastjærn. We choose to use this chatbot because it allows us to access users’ interactions with the chatbot. We could have chosen to investigate a different, more advanced and established chatbot, but since the data provided by the conversations are needed for us to further investigate the differences, we choose to use our own chatbot. If we had chosen to use a different chatbot, we would have to either ask the users to send us screenshots of the conversations or ask the owners of the chatbot to give us insight into their data. Furthermore, by using our own chatbot we gain more knowledge about what is needed to make a chatbot.

Our approach will be similar to experimental research, and we will organize it by dividing the participants into two groups and exposing each participant to only one condition (between-group design). The participants of each group will be aware of the existence of the association, but not necessarily know too much about it. This is because we want the conversations to be organic, and the questions to be genuine. The participants will be chosen at random, but due to practicalities, the first five participants will be directly assigned to the board member chat, and the last five participants will be directly assigned to the chatbot chat. Therefore, since the assigning of participants to conditions is not truly randomized, the experiment will only be a quasi-experiment.

We also considered talking with an expert on the theme at a later stage. As mentioned below, we have found articles about the effect of AI on language formality and politeness, but it would have been interesting to interview someone who works with this. However, we decided to focus on developing the chatbot further to get a more realistic evaluation. We will talk more about the challenge of this in the lessons learned chapter.

5.1. Gathering data

When gathering data, we will initially ask five students to chat with the Toastjærn association. They will be told that a member of the board is on the other side, and we want them to interact with them through our phones so that they remain anonymous. The students will be asked to ask the member about the association, and that our goal is to collect data on questions asked to the chatbot. The board member that is answering will look at the chatfuel page at the same time to try and give the same answers as the chatbot would give. This is to ensure that the answers about Toastjærn become as similar as possible to each group.

After that, we will ask five new students to chat with the chatbot ToastBot. In order to make sure that the data is comparable to the data gathered from the chat with the board member, we will ask them to do the same as the other group did (ask questions to the chat about the association).

5.2. Analysis of gathered data

When all ten people had chatted with either a board member or the chatbot, we analysed the data to look for similarities within the groups and differences between the groups. We tried our best to not make assumptions, but we already had our hypotheses in advance. This can have tainted the data and is something we have to take into considerations when we look back at the validity of our data.

To begin our analysis we took screenshots of all of the conversations and printed them all out. We also looked at the several user interactions the chatbot has had since it went live on the Toastjærn facebook page. We sorted them into two groups; talking to a board member, and talking to the chatbot. We individually read through all of the chats to see how the participants behaved within the groups, what was similar, what was different.

We went inductively into the data to see what patterns we could find. We looked at what kinds of questions they would ask and in what way. We also looked at the length of the conversations and in what way they would start and finish it.

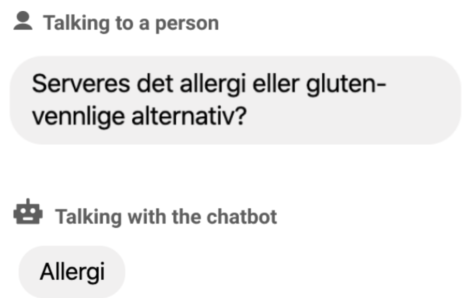
We then gathered each group members analysis and put them together to discuss them. Comparing the two groups we looked at our perception of the conversations and what our focus was in our analysis.

To analyse the differences between the two different groups we looked for similar questions. This was to see if the way the participants had asked their questions when talking to a person would differ from talking to our chatbot.
Lastly we categorized our findings into a report.

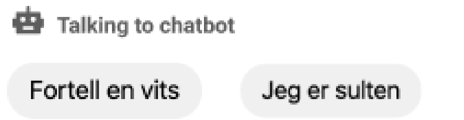
6. Findings

When talking to a person, our test participants were polite and thankful for the answers. Several of them used emojis, and they would write longer sentences describing their questions.
Sometimes, they even had follow-up questions after the initial answer.

When our test participants were asked to speak to a chatbot they were much more direct in their questions, often opting for one-worded indicators for their inquiries and leaving out punctuation marks.



Multiple of the users interacting with the chatbot tried “testing the limits” and trying to get the chatbot to answer funny questions or make jokes. None of the users tried doing this when they were talking to a real person.



When encountering errors with the chatbot, some participants quite quickly lost interest in conversing with it, while some tried to adapt and configure their questions to test if it would yield a new result.

Example 1:
This participant continues to feed single words into the chatbot, and when continuously receiving error messages in return, the participant quickly gave up.

Example 2:

This participant got error messages when asking questions, but was still curious to see if she could get her question answered by trying different ways of wording her questions.

Example 3:

Another participant encountered a false positive when asking about the price of a toast, getting the definition of a toast instead.

As shown in the title of this paper, one participant thought they were talking to a chatbot while actually talking to a human. A screenshot from the original chat is included on the front page, but here is a translated version of that conversation:

> P: What do you do?
>
> C: Hi :)
> C: We are a student association at IFI that has events where we eat toast together :)
>
> P: Sorry, I got the impression that this was a chatbot, I didn't mean to be so direct :'D
> P: Thanks for the answer!"

When we approached this participant, we asked them if they could ask some questions to the chat. Since we used the chat for the same association that we made the chatbot for in iteration 1 of this paper and this participant was aware of the existence of said chatbot, the participant was of the impression that they were talking to a chatbot and not a human.

As we can see in the example, the participant decided not to include elements like greetings, emoticons and explanations of their intentions for the conversation. When we reminded the participant that they were in fact chatting with a human, they felt sorry for the human on the other side and decided to express this in the chat.

7. Discussion

As mentioned in the section above, the participants in our test were often brief and kept their responses to one-word replies when chatting with the chatbot. This can be understood in light of Benotti and Blackburns (2006) theory of politeness with conversing with robots and conversational agents, where they emphasize both contextual information and the importance of understanding your conversational partners' desires and intent to prompt politeness (Benotti and Blackburn 2006). As we demonstrated in our study, there was a clear lack of contextual information, which might make the users seem quite disinterested and short in their responses.

The world's lead tech companies are moving from traditional graphical user interfaces to messaging platforms. (Følstad and Brandtzæg 2017, 38). In their research, Luger and Sellen found that when participants were learning to communicate with conversational agents they made use of a *particular economy of language* (2016, 5289). We found this to be true when the participants in our study believed they were chatting with a chatbot as well. Many of our participants dropped all pleasantries afforded to humans and essentially were looking for the most direct way to get answers to their questions, even when only suspecting that they were chatting with a chatbot. Patience for the system were overall quite low, where if errors occurred, most participants gave up quickly, while a few tried different configurations out of curiosity, to try to "make the system work". In Luger and Sellen's research (2016), employing a limited economy of words, where due to the discovered limitations of the system. In the case of technically advanced students at IFI, this practice came from having better mental models and familiarity with such systems. Could there be an overlap in these employed word economies that can be used to create better systems that appeal to a broader audience?

8. Conclusion

In this paper, we have researched the differences in user language when interacting with humans versus robots. We based this research on our hypothesis:

> *“Users are less formal when they know they are chatting with a robot compared to when they think they are chatting with a human.”*

Through between-group tests with 10 students at IFI and analysis of their conversations, we conclude that users are less formal when they know or think they are chatting with a robot, and therefore our hypothesis is confirmed. We believe that further research on how communication differ when users are messaging chatbots and humans is essential in order to create meaningful interactions, and that these findings should be taken into consideration when designing future chatbot dialogues.

However, it should be taken into consideration that the users we had were not a representative selection of the population as a whole. Our participants are students at IFI, and their technical knowledge may have affected the results of our study. Technically skilled users with better mental models of chatbots do not seem to want to interact the same way as they would with a human. Emulating interhuman conversations might therefore be the wrong approach for these users. On the other hand, GIFs and humor seemed to make for an overall better user experience based on the message dialogues and our observations.

If chatbots are to take a bigger role in everyday life in the future, more HCI research needs to be done regarding the expectations of different user groups. While the students at IFI did not address the chatbot in the same way they addressed the human, other users might be more inclined to address them the same way. This would be an interesting topic for further research on chatbot technology.

9. Lessons learned

By working with this paper throughout the semester, we have gained new knowledge about artificial intelligence. We have learned a lot about chatbots and robots in general and how to make and evaluate them, as well as how users interact with them and how to interpret that for further discussion.

When making the chatbot ToastBot, we learned that making a chatbot understand plain language can be a time-consuming challenge. We were presented with the option of making multiple choice answers that users could click instead of writing out their questions, but we wanted the chatbot to simulate normal written dialogue. We discovered that in order to do that, we needed to give the chatbot a lot of example sentences that we thought users might use if they wanted answers on a specific topic. When we first created the chatbot and tested it out with each other, we thought we had covered the most important example sentences. However, when we tested it out with the members of Toastjærn, we realised that we had missed several common wordings. Multiple of the members got error messages when asking for simple things like membership and events. We realised that we had to add more examples, so we ended up adding those questions the members asked, as well as some others.

After the first iteration, we decided to deactivate the chatbot until the next iteration. When we activated it again, we decided to test it before asking our participants to chat with it. This was a good call, because the chatbot did not understand anything we wrote to it. Due to this, we had to use a lot of time to make changes. We removed most of the examples and added them back in, and we also had to make some adjustments to the flow (what answers the bot was to give to certain buzzwords).

This experience taught us that making functional chatbots is an extensive task that requires a lot of work. While Chatfuel (the program used to make ToastBot) is not a very good or advanced chatbot template, it did actually (or perhaps obviously) need a lot of attention in terms of updates.

7. References

Benotti, Luciana, and Patrick Blackburn. 2016. “Polite Interactions with Robots.” In *Robophilosophy/TRANSOR*. (https://pdfs.semanticscholar.org/e78b/b318555d9dfd1507386b57eeb8c73a6bfa93.pdf?fbclid=IwAR24DLQC8dszV8sSe5GLpcQaPuiP8hf6dGPtsjEVTl6oH17WB4MaD6qBGpo).

Elgan, M. (2018). The case against teaching kids to be polite to Alexa. Fetched 22.10.19. https://www.fastcompany.com/40588020/the-case-against-teaching-kids-to-be-polite-to-alexa

Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of HCI. interactions, 24(4), 38-42. (https://dl.acm.org/citation.cfm?id=3085558)

Gupta, Swati, Marilyn A. Walker, and Daniela M. Romano. 2007. “How Rude Are You?: Evaluating Politeness and Affect in Interaction.” In *International Conference on Affective Computing and Intelligent Interaction*, Springer, 203–217.

Jenkins, M.-C., Churchill, R., Cox, S., & Smith, D. (2007). Analysis of User Interaction with Service Oriented Chatbot Systems. In J. A. Jacko (Ed.), *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments* (Vol. 4552, pp. 76–83). (https://doi.org/10.1007/978-3-540-73110-8_9)

Kocielnik, Rafal, Saleema Amershi, and Paul N. Bennett. 2019. “Will You Accept an Imperfect AI?: Exploring Designs for Adjusting End-User Expectations of AI Systems.” In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI ’19*, Glasgow, Scotland Uk: ACM Press, 1–14.

(http://dl.acm.org/citation.cfm?doid=3290605.3300641).

Luciana, Benotti, and Blackburn Patrick. ‘Polite Interactions with Robots’. *Frontiers in Artificial Intelligence and Applications*, 2016, 293–302. https://doi.org/10.3233/978-1-61499-708-5-293.

Luger, Ewa, and Abigail Sellen. 2016. "‘Like Having a Really Bad PA’: The Gulf between User Expectation and Experience of Conversational Agents." In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI ’16*, Santa Clara, California, USA: ACM Press, 5286–97. (http://dl.acm.org/citation.cfm?doid=2858036.2858288).

Needleman, R. (2017). Tech Etiquette: The Real Reason You Need to be Polite to Alexa. Fetched 22.10.19. https://callercallsback.com/tech-etiquette-the-real-reason-you-need-to-be-polite-to-alexa-6a76f97d0803

8. Appendix 1

Chatbot design task

The first part of our meeting was deciding on which case regarding chatbots we were interested in. We decided on making a chatbot for Toastjærn, a student association here at IFI devoted to the creation and consumption of toast. A major reason for this was that Inger Helene is an active part of the association, and saw that a chatbot could make their day-to-day communication with both potential members and current members more efficient.

Due to Inger Helene already being in Toastjærn and having access to previous chats, we were able to model our chatbot on existing data making our chatbot based on real cases. Through this data we made our chatbot quite extensive so that it still could be used after the assignment was done.
We did an informal user test at a Toastjærn event, where the participants were people in line waiting for toast. We tested three people, and got interesting feedback. We got insight into how people tested the limits of the chatbot as well as the threshold of errors before they gave up and did not use it anymore. We also observed that people were more impressed than we expected of the chatbot, and thought that it was exciting and fun.

9. Appendix 2

AI task

For module 2 we received a machine learning code from Dr. Morten Goodwin and dataset necessary to train a chatbot based on movie lines from the movie Gone with the wind.

Process:

Our process was one of trial and error. As novices to machine learning, and only having had one lecture on how to manipulate the data, we were quite confused in the beginning. We were not sure what parameters in the code to change or what to change it to. As Dr. Goodwin said during the lecture, there is no right or wrong answer. This is a try, test and evaluating process. We manipulated the number of iterations (epoch) in the code and the number of connections (dense) to some random values and reviews the results.

Outcome:

Test 1:

```
Train on 900 samples, validate on 100 samples
Epoch 1/2
900/900 [==============================] - 0s 115us/sample - loss: 3.2284 - acc: 0.1356 - val_loss: 3.6273 - val_acc: 0.0000e+00
Epoch 2/2
900/900 [==============================] - 0s 48us/sample - loss: 2.8095 - acc: 0.1722 - val_loss: 4.4602 - val_acc: 0.0000e+00
Finished training
ready
Chatbot:"I'm getting trashed, man." Isn't that what you're supposed to do at a party?
Human: No
Chatbot:What?
Human: Huh
Chatbot:Leave my sister alone.
Human: Leave MY sister alone
Chatbot:No -- something real.  Something no one else knows.
Human: Excuse me
Chatbot:Away.
Human: Rude
Chatbot:Something like that
Human: What do you want from me?
Chatbot:Who?
Human: ME
Chatbot:So he has this huge raging fit about Sarah Lawrence and insists that I go to his male-dominated, puking frat boy, number one golf team school. I have no sa
y at all.
Human: Honestly same
Chatbot:I hate him.
Human: Same
Chatbot:What?
Human: WHAT?!?!?!?
Chatbot:Now I do.  Back then, was a different story.
Human: Yeah, we all have lived lives before.
Chatbot:He just wants me to be someone I'm not.
Human: Oh, we are still talking about this?
Chatbot:Hey -- do you mind?
Human: Yeah, actually I do...
Chatbot:Is that right?
Human: YES
Chatbot:I guess I thought I was protecting you.
Human: From what?
Chatbot:If I was Bianca, it would be, "Any school you want, precious.  Don't forget your tiara."
Human: Now THATS a mood ;)
Chatbot:Oh, Bianca?  Can you get me my freshman yearbook?
Human: Huh?
Chatbot:I guess I thought I was protecting you.
Human:
```

Number of iterations (epoch): 2
Number of connections (dense): 512
Loss: 2.2284
Acc: 0.1356
val_loss: 4.4602
val_acc: 0
Number of interactions before crashing: Did not crash

Test 2:

```
Epoch 4/7
900/900 [==============================] - 0s 48us/sample - loss: 2.6443 - acc: 0.1722 - val_loss: 5.2759 - val_acc: 0.0000e+00
Epoch 5/7
900/900 [==============================] - 0s 47us/sample - loss: 2.6379 - acc: 0.1722 - val_loss: 5.4318 - val_acc: 0.0000e+00
Epoch 6/7
900/900 [==============================] - 0s 50us/sample - loss: 2.6337 - acc: 0.1722 - val_loss: 5.6276 - val_acc: 0.0000e+00
Epoch 7/7
900/900 [==============================] - 0s 50us/sample - loss: 2.6325 - acc: 0.1722 - val_loss: 5.7293 - val_acc: 0.0000e+00
Finished training
ready
Chatbot:Because I don't want to. It's a stupid tradition.
Human: What is?
Chatbot:Answer the question, Patrick
Human: My name is Bendik
Chatbot:Something like that
Human: Something like what?
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-3-e619be6a0d9d> in <module>
    104 s = " "
    105 while s:
--> 106     category = getCategory(s)
    107     text = getRandomTextFromIndex(category)
    108     print("Chatbot:" + text)

<ipython-input-3-e619be6a0d9d> in getCategory(inputString)
     89 def getCategory(inputString):
     90     #Get the correct classification
---> 91     token = tokenizer.sequences_to_matrix(np.array([makeTextIntoNumbers(inputString),makeTextIntoNumbers(x_train_org[0])]))
     92     aindex = np.argmax(model.predict(np.array([token[0]])))
     93     return aindex

/usr/local/lib/python3.6/dist-packages/keras_preprocessing/text.py in sequences_to_matrix(self, sequences, mode)
    410         x = np.zeros((len(sequences), num_words))
    411         for i, seq in enumerate(sequences):
--> 412             if not seq:
    413                 continue
    414             counts = defaultdict(int)

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```

Number of iterations (epoch): 7
Number of connections (dense): 512
Loss: 2.6325
Acc: 0.1722
val_loss: 5.7293
val_acc: 0
Number of interactions before crashing: 3

Test 3:

```
Train on 900 samples, validate on 100 samples
Epoch 1/7
900/900 [==============================] - 0s 129us/sample - loss: 3.3569 - acc: 0.1211 - val_loss: 3.4530 - val_acc: 0.0000e+00
Epoch 2/7
900/900 [==============================] - 0s 45us/sample - loss: 3.2312 - acc: 0.1722 - val_loss: 3.5447 - val_acc: 0.0000e+00
Epoch 3/7
900/900 [==============================] - 0s 44us/sample - loss: 3.0364 - acc: 0.1722 - val_loss: 3.7387 - val_acc: 0.0000e+00
Epoch 4/7
900/900 [==============================] - 0s 45us/sample - loss: 2.8307 - acc: 0.1722 - val_loss: 4.0636 - val_acc: 0.0000e+00
Epoch 5/7
900/900 [==============================] - 0s 42us/sample - loss: 2.7075 - acc: 0.1722 - val_loss: 4.4170 - val_acc: 0.0000e+00
Epoch 6/7
900/900 [==============================] - 0s 44us/sample - loss: 2.6646 - acc: 0.1722 - val_loss: 4.6938 - val_acc: 0.0000e+00
Epoch 7/7
900/900 [==============================] - 0s 46us/sample - loss: 2.6517 - acc: 0.1722 - val_loss: 4.8563 - val_acc: 0.0000e+00
Finished training
ready
Chatbot:Tell me something true.
Human: I love you.
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-4-025a1105c1ec> in <module>
    104 s = " "
    105 while s:
--> 106     category = getCategory(s)
    107     text = getRandomTextFromIndex(category)
    108     print("Chatbot:" + text)

<ipython-input-4-025a1105c1ec> in getCategory(inputString)
     89 def getCategory(inputString):
     90     #Get the correct classification
---> 91     token = tokenizer.sequences_to_matrix(np.array([makeTextIntoNumbers(inputString),makeTextIntoNumbers(x_train_org[0])]))
     92     aindex = np.argmax(model.predict(np.array([token[0]])))
     93     return aindex

/usr/local/lib/python3.6/dist-packages/keras_preprocessing/text.py in sequences_to_matrix(self, sequences, mode)
    410         x = np.zeros((len(sequences), num_words))
    411         for i, seq in enumerate(sequences):
--> 412             if not seq:
    413                 continue
    414             counts = defaultdict(int)

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```

Number of iterations (epoch): 7
Number of connections (dense): 86
Loss: 2.6517
Acc: 0.1722
val_loss: 5.7292
val_acc: 0
Number of HCI interactions before crashing: 3

Test 4:

```
900/900 [==============================] - 0s 81us/sample - loss: 2.6383 - acc: 0.1722 - val_loss: 6.1386 - val_acc: 0.0000e+00
Epoch 7/10
900/900 [==============================] - 0s 65us/sample - loss: 2.6367 - acc: 0.1611 - val_loss: 6.3117 - val_acc: 0.0000e+00
Epoch 8/10
900/900 [==============================] - 0s 65us/sample - loss: 2.6363 - acc: 0.1722 - val_loss: 6.5062 - val_acc: 0.0000e+00
Epoch 9/10
900/900 [==============================] - 0s 68us/sample - loss: 2.6393 - acc: 0.1667 - val_loss: 6.4885 - val_acc: 0.0000e+00
Epoch 10/10
900/900 [==============================] - 0s 65us/sample - loss: 2.6389 - acc: 0.1722 - val_loss: 6.4720 - val_acc: 0.0000e+00
Finished training
ready
Chatbot:I just need to lie down for awhile
Human: Me too.
Chatbot:Block E?
Human: Sure
Chatbot:Why 're you doing this?
Human: Because I am tired.
Chatbot:I guess I thought I was protecting you.
Human: Thanks, but no.
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-2-ed11121f4b09> in <module>
    104 s = " "
    105 while s:
--> 106     category = getCategory(s)
    107     text = getRandomTextFromIndex(category)
    108     print("Chatbot:" + text)

<ipython-input-2-ed11121f4b09> in getCategory(inputString)
     89 def getCategory(inputString):
     90     #Get the correct classification
---> 91     token = tokenizer.sequences_to_matrix(np.array([makeTextIntoNumbers(inputString),makeTextIntoNumbers(x_train_org[0])]))
     92     aindex = np.argmax(model.predict(np.array([token[0]])))
     93     return aindex

/usr/local/lib/python3.6/dist-packages/keras_preprocessing/text.py in sequences_to_matrix(self, sequences, mode)
    410         x = np.zeros((len(sequences), num_words))
    411         for i, seq in enumerate(sequences):
--> 412             if not seq:
    413                 continue
    414             counts = defaultdict(int)

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```

Number of iterations (epoch): 2
Number of connections (dense): 86
Loss: 2.6389
Acc: 0.1722
val_loss: 6.4720
val_acc: 0
Number of interactions before crashing: 4

Test 5:

```
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
Train on 900 samples, validate on 100 samples
Epoch 1/2
900/900 [==============================] - 0s 123us/sample - loss: 3.3891 - acc: 0.1489 - val_loss: 3.4099 - val_acc: 0.0000e+00
Epoch 2/2
900/900 [==============================] - 0s 45us/sample - loss: 3.3640 - acc: 0.1722 - val_loss: 3.4190 - val_acc: 0.0000e+00
Finished training
ready
Chatbot:Not all experiences are good, Bianca. You can't always trust the people you want to.
Human: This is true
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-4-8d4a9e2e20ac> in <module>
    104 s = " "
    105 while s:
--> 106     category = getCategory(s)
    107     text = getRandomTextFromIndex(category)
    108     print("Chatbot:" + text)

<ipython-input-4-8d4a9e2e20ac> in getCategory(inputString)
     89 def getCategory(inputString):
     90     #Get the correct classification
---> 91     token = tokenizer.sequences_to_matrix(np.array([makeTextIntoNumbers(inputString),makeTextIntoNumbers(x_train_org[0])]))
     92     aindex = np.argmax(model.predict(np.array([token[0]])))
     93     return aindex

/usr/local/lib/python3.6/dist-packages/keras_preprocessing/text.py in sequences_to_matrix(self, sequences, mode)
    410         x = np.zeros((len(sequences), num_words))
    411         for i, seq in enumerate(sequences):
--> 412             if not seq:
    413                 continue
    414             counts = defaultdict(int)

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```

Number of iterations (epoch): 10
Number of connections (dense): 2111
Loss: 3.3640
Acc: 0.1722
val_loss: 3.4190
val_acc: 0
Number of HCI interactions before crashing: 1

Test 6:

```
Epoch 297/300
900/900 [==============================] - 0s 43us/sample - loss: 2.6225 - acc: 0.1722 - val_loss: 9.6531 - val_acc: 0.0000e+00
Epoch 298/300
900/900 [==============================] - 0s 43us/sample - loss: 2.6234 - acc: 0.1722 - val_loss: 9.6849 - val_acc: 0.0000e+00
Epoch 299/300
900/900 [==============================] - 0s 41us/sample - loss: 2.6228 - acc: 0.1722 - val_loss: 9.7180 - val_acc: 0.0000e+00
Epoch 300/300
900/900 [==============================] - 0s 42us/sample - loss: 2.6231 - acc: 0.1722 - val_loss: 9.7461 - val_acc: 0.0000e+00
Finished training
ready
Chatbot:In 9th.  For a month
Human: What month?
Chatbot:You ' re looking at this from the wrong perspective.  We're making a statement.
Human: That is cool.
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-8-17005f256b69> in <module>
    104 s = " "
    105 while s:
--> 106     category = getCategory(s)
    107     text = getRandomTextFromIndex(category)
    108     print("Chatbot:" + text)

<ipython-input-8-17005f256b69> in getCategory(inputString)
     89 def getCategory(inputString):
     90     #Get the correct classification
---> 91     token = tokenizer.sequences_to_matrix(np.array([makeTextIntoNumbers(inputString),makeTextIntoNumbers(x_train_org[0])]))
     92     aindex = np.argmax(model.predict(np.array([token[0]])))
     93     return aindex

/usr/local/lib/python3.6/dist-packages/keras_preprocessing/text.py in sequences_to_matrix(self, sequences, mode)
    410         x = np.zeros((len(sequences), num_words))
    411         for i, seq in enumerate(sequences):
--> 412             if not seq:
    413                 continue
    414             counts = defaultdict(int)

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```

Number of iterations (epoch): 1
Number of connections (dense): 12
Loss: 2.6231
Acc: 0.1722
val_loss: 9.7461
val_acc: 0
Number of HCI interactions before crashing: 2

Test 7:

```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
Train on 900 samples, validate on 100 samples
Epoch 1/5
900/900 [==============================] - 0s 162us/sample - loss: 3.2198 - acc: 0.1600 - val_loss: 3.6584 - val_acc: 0.0000e+00
Epoch 2/5
900/900 [==============================] - 0s 49us/sample - loss: 2.7978 - acc: 0.1722 - val_loss: 4.5060 - val_acc: 0.0000e+00
Epoch 3/5
900/900 [==============================] - 0s 48us/sample - loss: 2.6594 - acc: 0.1722 - val_loss: 5.0755 - val_acc: 0.0000e+00
Epoch 4/5
900/900 [==============================] - 0s 49us/sample - loss: 2.6430 - acc: 0.1722 - val_loss: 5.2540 - val_acc: 0.0000e+00
Epoch 5/5
900/900 [==============================] - 0s 49us/sample - loss: 2.6372 - acc: 0.1722 - val_loss: 5.4632 - val_acc: 0.0000e+00
Finished training
ready
Chatbot:My father wouldn't approve of that that
Human: Approve of what?
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-9-c5aa4aa22c26> in <module>
    104 s = " "
    105 while s:
--> 106     category = getCategory(s)
    107     text = getRandomTextFromIndex(category)
    108     print("Chatbot:" + text)

<ipython-input-9-c5aa4aa22c26> in getCategory(inputString)
     89 def getCategory(inputString):
     90     #Get the correct classification
---> 91     token = tokenizer.sequences_to_matrix(np.array([makeTextIntoNumbers(inputString),makeTextIntoNumbers(x_train_org[0])]))
     92     aindex = np.argmax(model.predict(np.array([token[0]])))
     93     return aindex

/usr/local/lib/python3.6/dist-packages/keras_preprocessing/text.py in sequences_to_matrix(self, sequences, mode)
    410         x = np.zeros((len(sequences), num_words))
    411         for i, seq in enumerate(sequences):
--> 412             if not seq:
    413                 continue
    414             counts = defaultdict(int)

ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()
```

Number of iterations (epoch): 2
Number of connections (dense): 0
Loss: 2.6372
Acc: 0.1722
val_loss: 5.4632
val_acc: 0
Number of interactions before crashing: 1

Reflections and what we learned

Given the responses were based on movie lines, it was quite difficult to understand if the chatbot was just giving random responses or not, as it gave no indication as to why it chose the movieline it did as a response. It seemed to us that the more we wavered from Dr. Goodwin's initial parameters the faster the chatbot crashed, giving us a ValueError: *"The truth of an array with more than one element is ambiguous."*

10. Appendix 3

Evaluation of the Netflix recommendation engine

Subject, scope, what are you evaluating, why that system?
We are evaluating the recommendation engine of the streaming website Netflix. We are especially interested in how the recommendation functionality of the website works since we all have seen it “in action” and have had first-hand experience with it. Our assumption is that Netflix (as an AI-infused system) recommends series/movies based on use, and this we often have experienced ourselves when looking through the recommended section of other profiles than our own. Due to this, the scope of this evaluation will only be concerned with the recommendation functionality of Netflix and how this is influenced by the users’ activity.
We are all avid users of Netflix, and we use it almost daily. However, none of us have any relationship or understanding of how the recommendation functionality works beyond initial understanding that it recommends movies and series based on previous activity.

Plan for evaluating using guidelines for Human-AI interaction. How will you evaluate?
Our plan is to create two blank Netflix profiles, play movies and series and try to see how fast the AI recognizes the view patterns and recommend movies and/or series. After seven iterations of this, we will compare the “Recommended for you”-sections of both profiles.

We have decided to name the profiles ToastBot and HorrorBot: ToastBot will only be streaming food series, while HorrorBot will be streaming horror movies. Both profiles will be in English, not registered at children, and they will be connected to the same account. We will stream food series on ToastBot and horror movies on HorrorBot simultaneously, and we have decided that we will continue watching new episodes on ToastBot until HorrorBot is done streaming the movie.

Every time HorroBot finishes a movie, we will exit both screens and go to the Netflix homepage of both profiles. Since new profiles do not have the "Recommended for you"-section yet, we will have to look for it the first time after we finish a horror movie. The following iterations, we will look at how the "Recommended for you"-sections differ from the last time. If the recommended sections do have relevant elements in them, we will choose to stream those elements. If the sections do not recommend the right type of elements, we will use the search engine to find more fitting options. If so, we will only use the search words "food" and "horror".

When we compared the recommended sections with their previous selections and each other's selections, we will empty the view history and see if there are any changes to the recommended section now.

Guidelines for interaction - Netflix

Netflix does well:

G4	**Show contextually relevant information.** Display information relevant to the user's current task and environment.

You can not only search for titles of movies or shows but also the names of the people who star in the shows and get all show this actor/actress appears.

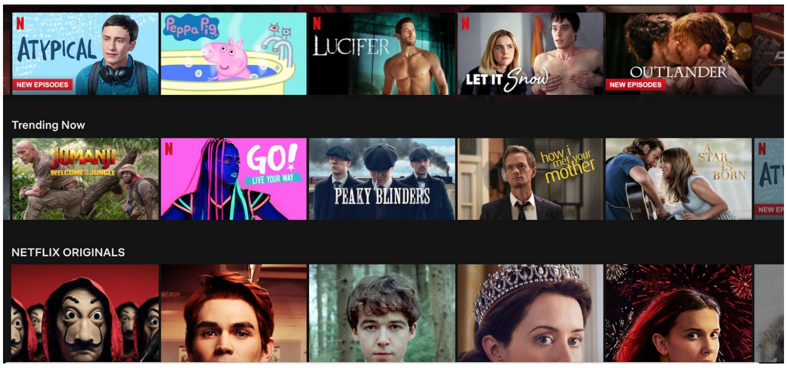
Also if you search for a show which Netflix does not offer, it will recommend other shows available based on the user input.

G13	**Learn from user behavior.** Personalize the user's experience by learning from their actions over time.

Results:

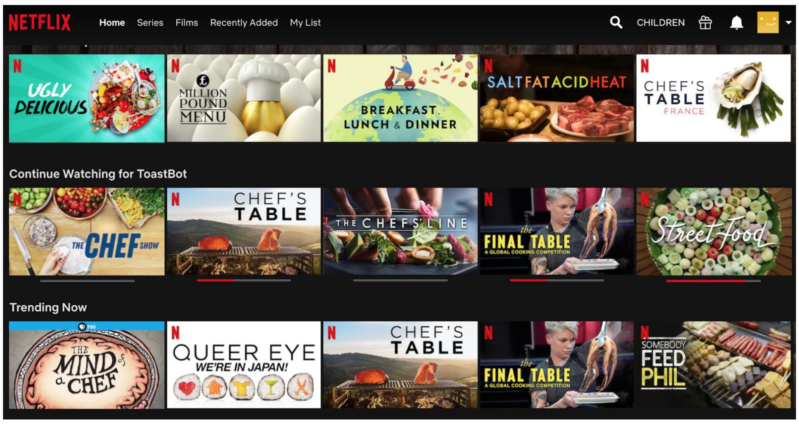
After one week of running Netflix on the two profiles we created, the two profiles looked drastically different. Both G4 and G13 were clearly used as shown in the screenshots from the two profiles below:

This is how the profiles looked before we started to watch very specific movies and shows:



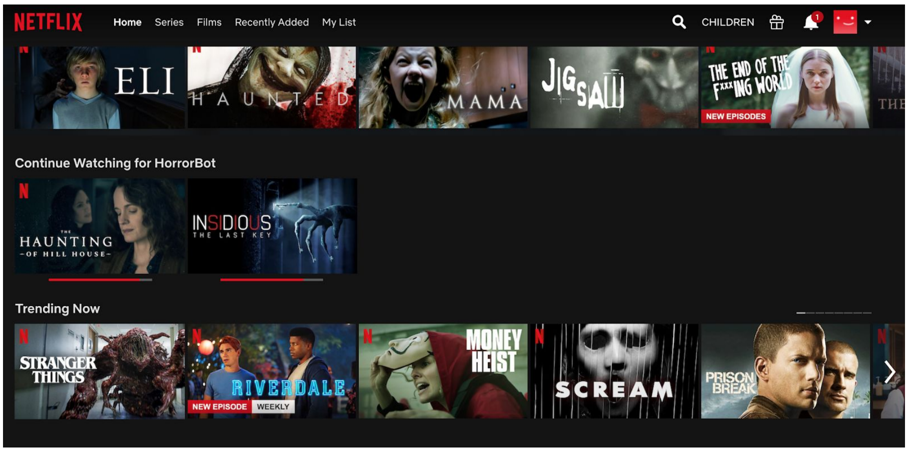
Whereas after highly specified “watching” the two main screens of Netflix appear very different:

ToastBot (the one profile only watching cooking shows):



Here we can clearly see how Netflix have tailored the recommendations of the specific users, as almost all the “trending” shows are cooking shows. The same goes for the first row of shows presented to the user.

HorrorBot:



For this profile we also see a string leaning towards the horror genre. Like ToastBot, this becomes apparent in the shows and movies presented to the user. There is a clear leaning towards the more “dark” horror and thriller-like shows.

Netflix does support, but could improve:

G8	**Support efficient dismissal.** Make it easy to dismiss or ignore undesired AI system services.

After completing a show, you can easily give feedback if you enjoyed it or not, using a thumbs up/down approval system. Which affects how your recommended shows appear and also gives you a percentage of how much of a “match” a show or movie is on the background of your viewing activity.

G9	**Support efficient correction.** Make it easy to edit, refine, or recover when the AI system is wrong.

On Netflix, users are able to dismiss recommendations, i.e in this context removing recommendations based on undesired material. E.g say someone else watched their favorite show on your account or similar, and you do not want recommendations based on this input.

Netflix does support these actions, but they are limited to the browser version, as this is not supported in the app versions.

After completing the test, we removed all viewing history. There was no clear change in the recommendations given by Netflix from both before and after we hid the viewing activity, making us unsure if there had been a change or not.

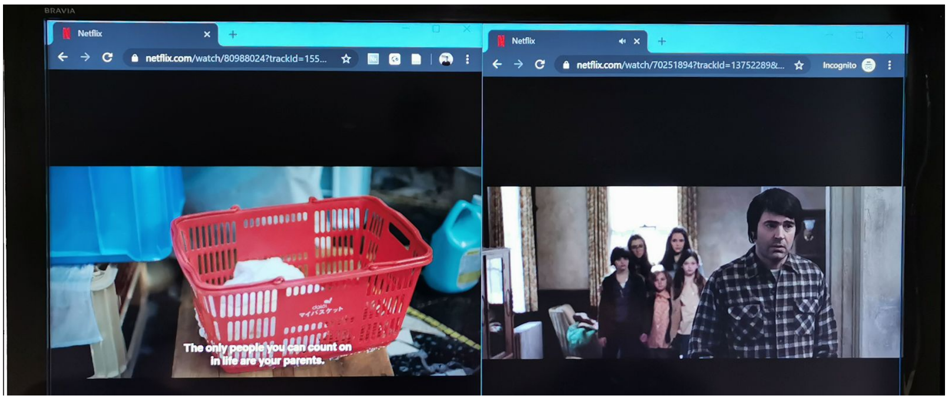
Process

When we created the new profiles, we were asked to choose three movies or series that we liked to make the AI make recommendations for us. We first tried to find three food shows, but since the selection, Netflix provided did not include any food shows, we decided to skip this for both the ToastBot and the HorrorBot. When we then entered the home screen, both profiles presented the same categories, as well as the same movies and series within them.

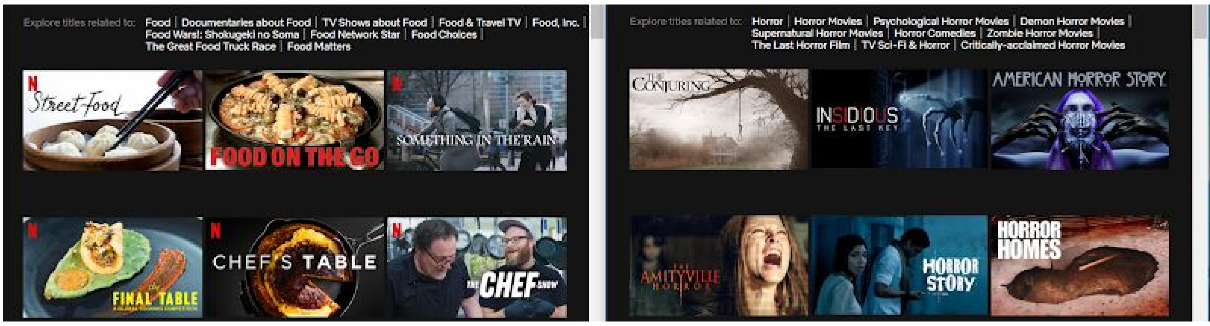
However, when we tried to open a new profile on a different account, some of the movies and series have switched places, but most of them were present in both the tested profiles and the control profile.

(A screenshot of the ToastBot home screen before watching or searching for anything. The home screen of HorrorBot was identical.)

We decided to use the search engine to find the right movies and series for the two profiles. In ToastBot, we searched for “food” and chose the first element, “Street food”. It was coincidentally a series. In HorrorBot, we searched for “horror” and also chose the first element, “The Conjuring”. This was coincidentally a movie. We split the screen to watch both streams at the same time, and let them run in the background while we wrote on the discussion part of this essay. This allowed us to pay attention to unforeseen interruptions to the streams, to requests to confirm that we were still watching, and to the end of a stream.



When The Conjuring ended, Netflix recommended we should watch “The King” straight after. Its trailer started playing automatically. We stopped the trailer, stopped the stream on the ToastBot profile and were sent back to the search page where we first found our first stream elements:

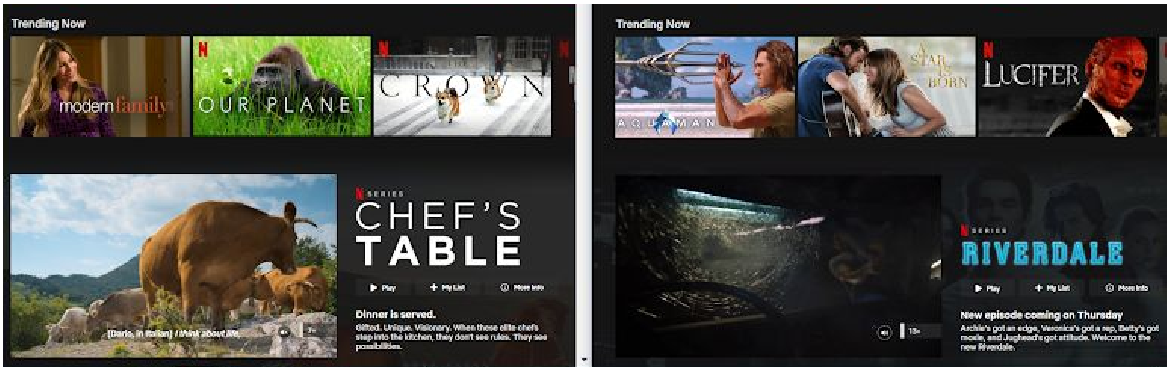


When we entered these pages the first time, we had not watched anything yet and therefore we did not get any predicted matches with any of them. At first, we thought this was still the case, because we only checked ToastBot’s page first.

However, when we checked HorroBot's page we saw that it actually did get an estimated percentage of the match between the profile owner's preferences and other movies and series (here Insidious: The Last Key).

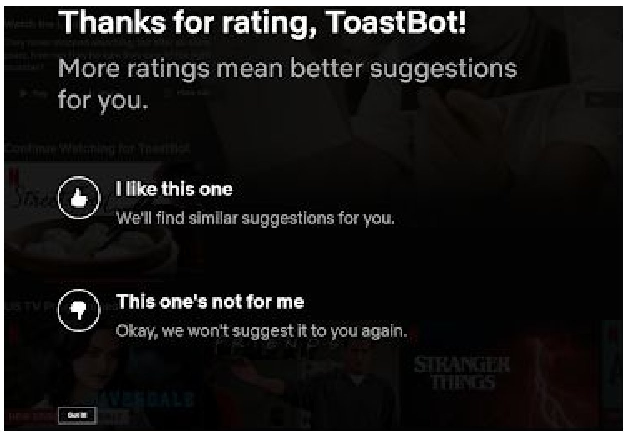
We also observed that there was no change to the ratio of movies vs. series in either profile since neither of them were changed.

We decided to go back to the home screen to check out the recommended section. To our surprise, it did not exist yet. This might be because each profile only watched one show or movie, which made it hard for the algorithm to suggest anything yet. What we did see was that the home screens now differed slightly. While Chef's Table was highlighted for ToastBot, Riverdale was highlighted for HorrorBot.



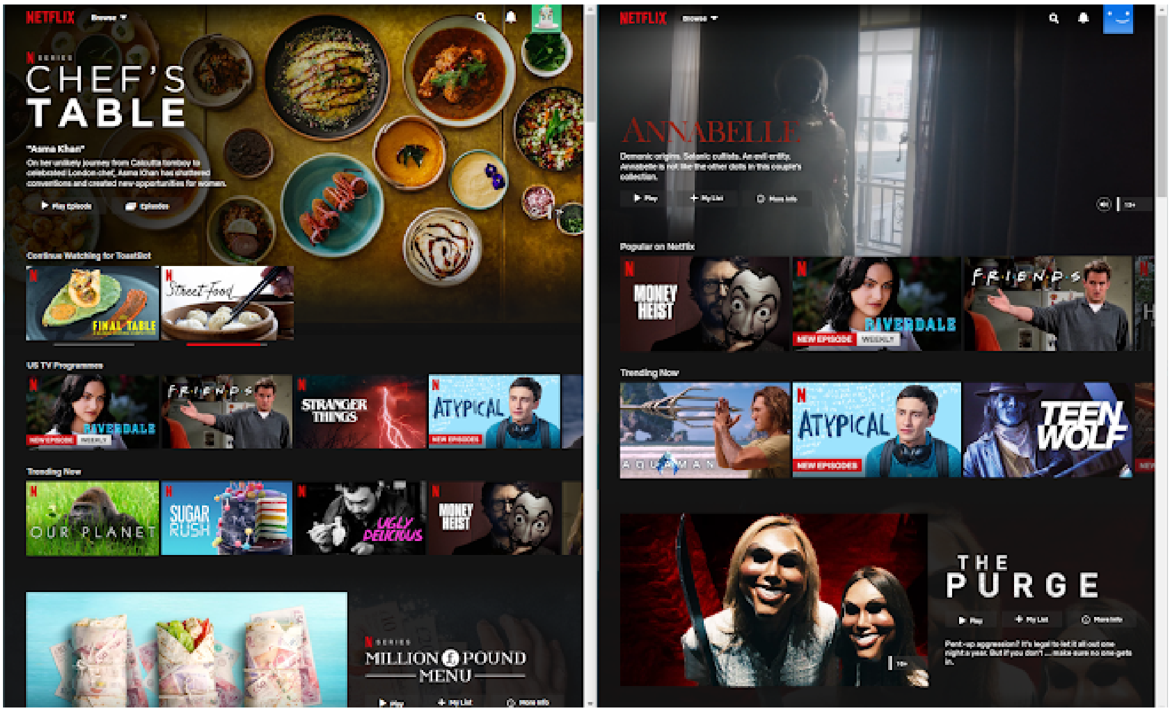
We also observed that the category "Horror movies" was higher up on HorrorBot's page than on ToastBot's page, but neither has a "Food" category yet.

We went back to the movies and series we had just watched and liked them. Netflix made it clear to us what the actions of liking and disliking would do (Find similar suggestions vs. won't suggest [the element] again).



Since neither profile had a recommended section yet, we decided to search for "food" and "horror" again. It turned out that the page that showed up was the same page that showed up. We chose the next elements to fulfill our criteria: The Final Table for ToastBot, and Insidious: The Last Key for HorrorBot.

When Two episodes of The Final Table and the movie Insidious: The Last Key were done, we went back to the home screen. There was still no "Recommended for you"-section, but the AI had started to pick up on some patterns. While the two series Chef's Table and Million Pound Menu were highlighted for ToastBot, the two



movies Annabelle and The Purge were highlighted for HorrorBot.
Since Chef's Table and Annabelle both fulfilled our criteria, we decided to stream them next. The episode Netflix recommended was the third episode of season 6. We decided to start on season 1 episode 1 to more accurately simulate normal user activity.

At this point in our evaluation, the owner of the Netflix account deleted all the accounts because the group member who shared the account with the owner did not tell them that they were doing a school project with the Netflix account. The owner has a history of being hacked and was afraid that they had been hacked when they saw the extra accounts and the names “TestBot”, “ToastBot” and “HorrorBot”. When you delete a Netflix profile, you also delete the view history and all other data connected to that profile. This means that the data we wanted to delete at the final stage of our evaluation is already gone.

In order to not let our work go to waste, we wanted to try to replicate what we had already done. A new group member took upon them the responsibility of making the accounts and replicate the user activity. By doing that, we wanted to see if we would get the same recommendations or not.

Results and lessons learned:

As noted over, we saw a clear change in the two profiles after seven iterations of highly specific movies and series after many hours of watching specific movies and shows. This was very much in line with what we expected to happen.

Through this assignment we learnt that the AI-infused system of Netflix recommendations gave highly specific recommendations after we gave it highly specific data to work with. As mentioned over, we did not really know much about how Netflix choose the recommendations it did and viewed it as a “black box”. After interacting with it we now have some more insight into how this takes place. By choosing to “like” or “dislike” a movie og TV show, we get feedback to whether netflix will find more shows like this or not. The process of finding new entertainment is still somewhat unclear to us. Thus **G16** is somewhat followed, as we as users get some insight into how our actions will impact the system.

G16	**Convey the consequences of user actions.** Immediately update or convey how user actions will impact future behaviors of the AI system.

Netflix does not explicitly notify users of change, it more or less just “does it”, thus not following **G18** that well. This could be something for Netflix to improve. By doing so, the user will understand how and why a certain movie or show is recommended based on the users activities.

G18	**Notify users about changes.** Inform the user when the AI system adds or updates its capabilities.

The guidelines for interacting with an AI-infused system have given us explicit ways of evaluating the system. They have been useful for us to be able to talk about both advantages and disadvantages of the system we evaluated. The guidelines have been a useful framework for us to evaluate the system and to give specific recommendations for improvement. They have showed us the benefit of having an explicit way of evaluating a system like Netflix, as these types of AI-infused systems tends to be somewhat abstract and intangible.

11. Feedback from iteration 1

The feedback we got was mostly positive, though two main points we needed to take into consideration were mentioned.
First being we might not get enough data only collecting data from the ToastJærn chatbot. Second, how we were going to get realistic data during our tests, as the behavior of the participants will most likely be influenced by the fact that they are being observed.

For the first point of concern, we agree that our small data collection is not enough to make any true assumptions about our research question, but we think it will be enough for the purpose and scope of this assignment, to see if we can find any initial emerging patterns.

To try to get realistic data, we did not watch them as they interacted with the chatbot as the participants engaged with the chatbot, but our presence did still probably have an effect. But we did not only use data from our tests. The chatbot has been operational since the start of September and has had several user interactions since then. We were not present for any of these interactions, having no effect on the users. This data was also used in our findings.