Report on Software Engineering and Computer Science

Lynna Xie

Introduction

In this report, I wish to demonstrate my understanding of software engineering and artificial intelligence. My understanding comes from class activities, a field trip to a local IT company, teacher notes, the computer science field guide, and other online resources.

Software Engineering

Introduction

Software engineering is the application of engineering to the development of software in a systematic method. Software engineering is a huge and complex project, which involves lots of phases to develop the whole project. Usually, the phases include analysis, design, implementation, and testing. The outcome of software engineering is the result of cohesion and teamwork. In this topic, I will be evaluating the problems that make the projects fail, basically poor communication and complexity, and exploring the effective solutions for those problems. We are aiming to decrease the rate of failure during the process of software engineering.

Why software projects fail.

Poor communication is the reason why most IT projects fail, according to a Web poll released by the Computing Technology Industry Association. There is lots of communication between parts involved in one project, such as how the customer explains, how the project leader understands, how the analysist designs, how the programmer writes, how the business consultant describes, and so on. Any time that one party misunderstands or misconstrues the words or actions of another, it has the potential to create conflict. Poor communication is frustrating in the workplace and can lead to poor performance, lack of teamwork, low morale, and reduced profits. What is worse, if there is a misunderstanding at the beginning of the software development projects, the whole project can fail.

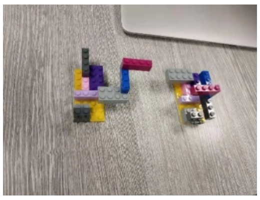
Complexity refers to the complex properties of a piece of software, all of which affect internal interactions. With the increase of complexity in software development, it becomes nearly impossible to know and understand all the parts inside. Similarly, higher level of complexity in software increases the risk of unintentionally interfering with interactions and so increases the chance of introducing defects when making changes. In more extreme cases, it can make modifying the software virtually impossible. Regardless of poor communication, complexity itself has been a barrier to success. With the influence of communication, complexity makes the projects' goals even hard to achieve.

In 2018, doctors and hospital staffs of the Wales NHS experienced a widespread computer failure that led to them being unable to access patient files. According to the National Cyber Security Center, the failure was due to technical issues as opposed to a cyber-attack, yet it still caused wide disruption as GPs were unable to access blood and X-Ray results. It also caused a backlog as patients could not be contacted to cancel appointments, and notes could not be typed up and saved on NHS systems.

School Activities (Building Block)

Today in class, we imitated building a project with only verbal communication. My friend and I tried to build the same combination using two identical sets of Lego blocks. She gave me instructions on how to build her blocks and I followed. However, though we have a good tacit understanding and tried our best to figure out the way of connection. We still came up with two slightly different projects.

Conclusion of the activity



This activity shows that poor communication is an important issue, and it occurs in every team project. I misunderstood some of her descriptions, and other of them were not specific enough. When we received information, we had our own understanding, but it became a barrier to effective communication. In software engineering, projects can only be more difficult and complex, which leads to a higher probability of misunderstanding at any stage. When an error occurs, it has a ripple effect that affects the entire project and subsequent work.

The four stages while developing engineering projects.

The four main stages while developing engineering projects clearly defined and distinct work phases, which include analysis, design, implementation, and testing. The stages are also known as part of the system development life cycle. The cycle aims to produce high-quality systems that meet or exceed customers’ expectations, based on their requirements.

Analysis

Analysis is the first stages of the whole development cycle. It is the blueprint of the whole project, which determines how to carry out the process and suggests the prospective opportunities and directives, as well as analyzing the situation and project goals. In this stage, developers will cooperate with clients to have a further understanding of their requirements and discuss the details to avoid misconceptions between developers and clients. More than communicating with clients, there needs to be efficient communication between designers and developers to achieve the information maximization and ensure to produce a fitting project in later stages.

During my assessment of coding the Python program. The analysis part is reading the instructions and the requirements and thinking about the structure of my whole program. Sometimes I discussed with my teacher to ensure my project is on track. Also, I highlighted some important points that I needed to meet so that I would concern with them when I was doing the design.

Design

In the design stage, the functions and operations should be described in detail, including screen layouts, business rules, process diagrams, and other documentation. The output of this stage will describe the new system as a collection of modules or subsystems. The whole project is extremely complex. Designers need to break a complex project into smaller tasks, so that engineers can proceed step by step, thus reducing the possibility of mistakes. If the project is not effectively decomposed, it will be difficult to figure out what goes wrong once the warning is given.

When I was developing the Python program before, I treated it as a composition of several fragments instead of one whole program. I divided it into different functions and classes, and each class consists of several variables. I also sketched the interface layout before I started to code the program. These steps allowed me to have a clearer pathway and improve efficiency when developing the projects.

Implementation

After the design stages, it is time to carry out implementation. It is the process of putting a decision or plan (design) into effect it is the coding part in software development. It is the longest phase throughout the whole project as it is the core stage that determines if a project can be brought into life. Due to the property of complexity, the implementation stage is broken down to several tasks that allocate to different engineers. Therefore, good communication is required to connect each separated part into a whole. Otherwise, these parts may be hard to be connected, which finally lead to projects to failure.

For me when developing the website for the assessment, I used Notepad ++ as a tool to code HTML, CSS, and PHP. Implementation, in this case, was coding for the layout of the website I have drawn on a paper, to the actual website that everyone can access on the Internet.

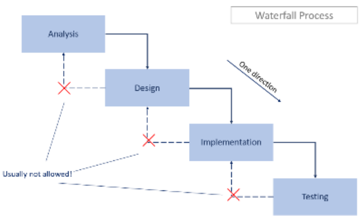
Testing

Testing is the fourth and final phase of the development cycle. Once the software is complete, it will be deployed to the test environment. The test team began testing the functionality of the entire system to verify that the entire application worked according to customer requirements. If there are bugs that indicate that the software is not working as you expected, the program needs to be returned to the developer for repair then send back to test team to re-test it. This process continues until the software is bug-free, stable, and ready to work according to customer requirements for the system.

For me when putting the website into testing, the layout usually not comes out correctly for the first time while a bunch of warnings appears. Thanks to the testing stage I get a chance to fix them all before I hand in. Usually, it is because of the spelling of the words in my codes and the misestimation of width and length that make my layout not come out as I hoped. After testing and re-testing and fixing the bugs following the warnings, the website finally loads correctly.

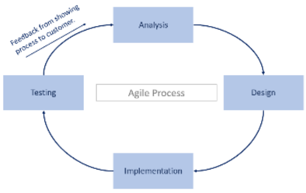
Waterfall Process

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of analysis, design, implementation, and testing. In the waterfall process, it is not allowed to go back to the last phase to do refinement.



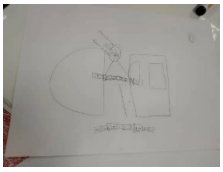
In the olden days, Waterfall model was used to develop enterprise applications like Customer Relationship Management (CRM) systems, Human Resource Management Systems (HRMS), Supply Chain Management Systems, Inventory Management Systems, Point of Sales (POS) systems for Retail chains etc.

Agile Process



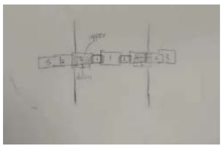
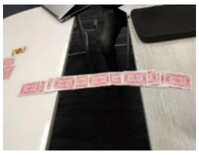
Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams. Different from the waterfall process, it allows developers to go back to the last stage to refine the plans and indications. Its value is derived from collaboration combined with early and continuous delivery.

School Activity (Bridge Making)



To understand agile and the waterfall methods better, we did some activities about waterfall and agile in class. We used cards as our basic material and built a bridge (about 30 cm) between our desks. One student in our group will be a designer, and the rest of us will be bridge builders. First, we tried the waterfall method to see if it will work. After the designer gave us the design drawing (the paper above), we then started to build the bridge(implementation) exactly following the design. For example, the base of each side is designed to be 5 cards, then we could only use 5 cards as our base while building. However, we failed in this attempt, the bridge fell like that (the picture below).

In this case, we have failed because the waterfall method does not allow people to go back to the last step to redesign so waterfall method is not the best way to build this project. Therefore, we changed our method to agile. We told her that the middle card should have a width instead of a length. What is more, there should be pressure on either side of the card to make the link stronger. Then she came up with a second design.

Sadly, we still failed this time. The reason might be that there were not enough cards to pressure the base and the central part of the bridge is not strong enough. We asked her to improve the design again.
Then we came up with third design drawing. Luckily, we finally succeeded!

Conclusion of the activity

From this activity, we learned that agile is a better approach to software engineering than waterfall. It allows people to go back to the last step (design) to develop errors, which allows us to communicate better across the team. It is not a very complex project because it generally consists of only two phases, design, and implementation. However, we still need to pay attention to the quality and hardness of the card and other factors. We are not architecture students, so we cannot immediately calculate and estimate the correct structure. When we are developing in an unknown or unfamiliar area, agile methods allow us to gain knowledge by trying to implement it. However, for some experienced or professional people, they can immediately use the waterfall method as their design work to improve efficiency.

Comparison

The waterfall method is an old school method. As projects become more complex, there is not enough flexibility to adapt to technological developments. Agile provides a more flexible approach to development. As projects become more complex, it is common to make mistakes during development. Agile allows them to exist from the last step and gives them the opportunity to return and fix these errors. Therefore, it can reduce the impact of complexity. For customers, they may not know clearly about their requirements before they see the software available, so they may change requirements, leading a series of refinement, which increases costs. In a long and complex project, time, money, and resources will be wasted if customers find that the final project is not what they expect. In agile, the team will give phased results to get feedback from clients so teams can adapt better to change and respond earlier. Therefore, changes are small and gradual, rather than sudden big changes. As a result, they can test programs constantly to detect and fix problems earlier in the whole process.

For me when developing the assessment of coding in Python I was using the Agile method. I sketched of the layout in advance, which are widgets and grids within the frames. However, I found there should be more space between different sections, so I developed my sketch. Also, I planned the classes and functions that I may use before I started to implement, but there were still some logic links that I did not concern at first, so I added a lot of new functions during coding. Using agile in this case was better as it allowed me to come up with new ideas to improve my projects along developing.

Although agile is popular in software development today, it still inevitably has some drawbacks. Since agile is about refinement by getting feedback, developers and customers must constantly communicate to ensure the project is on track, which costs developers lots of time. After several changes, the document may deviate from the first draft, so it's difficult for new members to catch up with the progress. However, if there is a complete working design document (which is the intent of large up-front designs and waterfall models), new team members or even new teams can familiarize themselves by reading the documentation.

Unlike software projects, in civil engineering, the construction of structures is irreversible. In this case, the waterfall would be a better choice because it is linear and can save a lot of money and avoid rebuilding it. They usually build a demonstration model before implementation so they can verify that the design works.

In agile, measuring progress is more difficult than waterfall, because progress occurs in multiple cycles. There is no clear endpoint, so projects can go on forever and incur additional immeasurable costs. The waterfall model provides a structured approach; The model itself is easy to understand because it develops linearly through discrete, understandable, and interpretable phases. This model is used only when the requirements are clear and fixed.

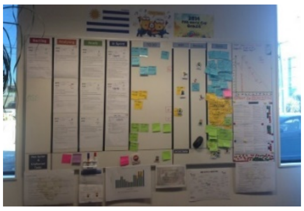
Trip to Orion Health

Our class got a great opportunity to visit Orion Health in Grafton, which is a global company that helps develop software for healthcare providers. We learnt the scrum methodology from a presentation by one of the speakers and had gained a lot by talking to those more experienced people.

At the end of the trip, we did an agile game which is about tennis balls. In the first round, we estimated 20 points and got 30 points but in the last round, we got 144 points. The success was from the retrospective in each round. We learnt from the sprint(implementation) and fixed the bugs we had then improved. In the first round, we used 1 ball but at last, we used 4 and did 2 cycles at the same time. Good communication is important because when everyone plays a role working out the projects, they need to have a good understanding of their task. We could tell from the activity that agile is a good way in developing software because it allows improving plans from practical experience.

Scrum Method

Scrum method is a form to achieve agile methodology and follows iterations through sprints. In scrum, the whole project is broken into several small tasks and they are allocated to each team member. A sprint contains a set of goals which must be completed by a specific date that is never longer than four weeks. A whole scrum includes 5 components which are backlog, sprint planning, stand-up, demo, and retrospective.



By Orion Health

During the tour in the company, we can feel that the atmosphere is open and the work is flexible. People were standing in a circle and giving out their opinions. From the talk, we knew that it's the 'stand-up' meeting. Every teammate needs to expound what he did yesterday, what he will do today and what impediments he has met. From the tour, we have a preliminary understanding of how a software development company works and how they used scrum(agile) to produce better communication between groups and between teammates.

Developers can provide clients with partial projects (demo) to check if the project is going along what they expected. This is an advantage of scrum which can significantly deal with the poor communication between clients and developers because they have more chance to correct their projects. Breaking down the big projects into small pieces of tasks is also breaking down the complexity. Looking for and fixing up the errors in small tasks are much easier than handling the whole big one. Agile methods are helpful when dealing with complexity and poor communication in software engineering projects.

Artificial Intelligence

Introduction

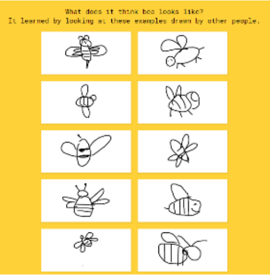
Artificial intelligence (AI) is a branch of computer science, which is the reason that I am going to talk about it in this report. There are many definitions of what AI is, but in my own words, it aims to understand the essence of intelligence and produce a new kind of intelligent machine that can react in a similar way to human intelligence. AI can perceive something happens in its environment and acting in intelligent ways to help achieve specific goals. Like the driverless cars, staff-less shops and even the sweeping robots are all examples of Artificial intelligence projects. What they have in common is that they all try to imitate human intelligence. This can help liberate human labor and work more efficiently.

A famous example of Artificial Intelligence is AlphaGo, developed by Google. It is a computer program that plays the board game Go. At the 2017 Future of Go Summit, its successor AlphaGo Master beat Ke Jie, the world No.1 ranked player at the time.

AlphaGo and its successors use a method called Monte Carlo tree search algorithm to 'calculate' its moves based on the 'knowledge' previously gained by machine learning, specifically by extensive training using an artificial neural network. The neural network can automatically generate identifying characteristic from the learning source. Therefore, it can predict AlphaGo's move that most likely to win.

On 11 December 2017, DeepMind released AlphaGo teaching tool on its website to analyze winning rates of different Go openings as calculated by AlphaGo Master. The teaching tool collects 6,000 Go openings from 230,000 human games each analyzed with 10,000,000 simulations by AlphaGo Master. Therefore, this AI has been deeply developed as it can easily analyze every situation from this step and draw out the best moves. AlphaGo is now unbeatable in Blitz Go games because human brain is not able to calculate as fast as AI to figure out the best step in a short time and multitask like AI to concern everything on the board. Artificial Intelligence has been booming developing and can replace human beings in some cases.

The victory of AlphaGo winning is a milestone in artificial intelligence research. Go has been regarded as a hard problem in machine learning and was expected to be out of reach of the technology of that time. As I mentioned, DeepMind used methods like machine learning and neural network to help develop AlphaGo, which are also effective methods in other AI development. An example of machine learning is picture recognition, we want the AI to recognize if one picture contains a dog or not. Millions of pictures will be provided and tagged as 'There is a dog' or 'There is not a dog', and the AI is required to train itself to recognize the feature and then successfully estimate if there is a dog in other pictures. Machine learning is usually used to train the neural network. A neural network includes several layers of executive program, where the information is estimated and passed to the next layer. It's based on how we think a human brain works; different levels of neurons involved in thinking through a task.


What does it think bee looks like?
It learned by looking at these examples drawn by other people.

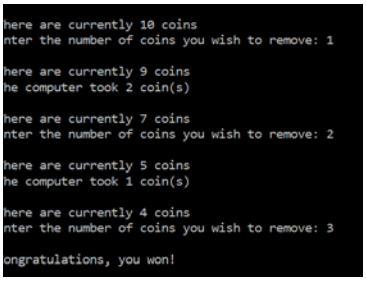
What is more, another example for the neural network is the 'Quick, Draw!" developed by Google. It challenges players to draw an object and then use the neural network to recognize what the player has drawn. When we are playing, the program is also collecting our drawing to expand its database and it helps to collect more features of a particular object. The picture on the left shows several drawings from different persons for the item 'bee'. AI can, therefore, recognize what it is when it has collected enough drawings. Just like we are human, we can recognize those pictures are bees because we have seen the real bees and know there are wings, stripes and sometimes legs and sting, which are the common features. However, most people cannot recognize other Apoidea such as Philanthidae, Mellinidae as zoologists can. This is because we do not know the features of those Apoidea. Neural network in AI works in almost the same way, that they can only recognize what they have been trained to recognize.

However, there are still problems in AI research. The AIs I mentioned before are all too 'specialized'. Like the AlphaGo, it is undoubtedly an expert in Go, but it cannot recognize the pictures of 'bee'. Not like a real human being, we can play Go games and recognize a bee. The key problem here is the pre-set rules for AI. AIs can only be triggered by situations. Like the image recognition one I mentioned before, it may be able to recognize if there is a dog after being trained by millions of dog's pictures, but if we give it a cat picture, it may still recognize it as a dog as their features are similar. Or like Siri AI chatbot, can only respond when being input something. For example, when we ask for the weather forecast, it searches up and displays the weather. They will not start chatting with us as a human does. What is more, Siri can only respond on simple questions and when we ask complicated questions, it will only say 'Sorry, I don't understand'. AI can only respond to questions that have been set-up to reply. Because of the limitation of its database, there are limited kinds of response. Comparing to human, we can always answer differently for various questions. The lack of flexibility of AI is, therefore, a key problem which is due to the pre-set rules.

Another problem in AI research is the inability to learn. AIs can only react to a certain condition and just directly provide the solutions. However, the do not understand the concept or reason behind the solutions. Like the AlphaGo, it only knows how to play---what is the best next move from the statistic. It cannot understand the underlying reasons why it should put down that chess. It does not possess general-purpose intelligence but only for the Go game. It cannot act like a human, who can wake up someday and say, 'Today I'm going to study the difference between different kinds of Apoidea because I suddenly become interested in it.' AI can just process to start recognizing but can never (at least for now) 'come up with' an idea to do something. However, sometimes it can be an advantage as it helps reduce the emotional effects on the judgments.

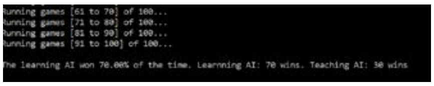
Reinforcement Learning

Reinforcement learning is one of the methods that AI can use to deal with the problem of an inability to learn. By using reinforcement learning, AI can learn from its mistakes and remember the errors situations. If there is a complicated maze with several blind alleys. When AI meets the end of the road, it then knows that it is the wrong way to the exit so AI can know it is the wrong way. Next time when coming up to the same road, it can avoid the mistake again. This is how the reinforcement learning works, that is, AI learns from the mistakes we have made and remember, therefore it will take correct action next time.

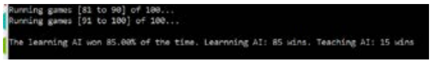


```
here are currently 10 coins
nter the number of coins you wish to remove: 1

here are currently 9 coins
he computer took 2 coin(s)

here are currently 7 coins
nter the number of coins you wish to remove: 2

here are currently 5 coins
he computer took 1 coin(s)

here are currently 4 coins
nter the number of coins you wish to remove: 3

ongratulations, you won!
```

An example of helping understand how the reinforcement learning works is in an activity we did in our class. It is a popular game and works by, starting with 10 coins, players involved (human and computer) can remove up to 3 coins each time, the one who removes the last coin loses. Before letting the AI learn, I played with the 'ignorant' AI. From the picture, we can see that it was easy to win --- I just input consecutive numbers 1,2,3. The program picked a random number, following the rules to finish but not to win the game. We can tell that this AI was not intelligent at that moment. However, the program can learn through reinforcement so playing more games would help increase its intelligence. As the picture shows below, we can see the victory rate gradually increasing as more games were played. Without learning, the rate is just only 50%. This means the AI was becoming more and more intelligent. It learns from the games that it has played and remembered the moves of winning. After learning from several games, the AI becomes more unbeatable and closer to human intelligence as it starts to learn like a human being. However, the shortage of AI, in this case, is that it can only learn from what happened but can't predict



```
unning games [61 to 70] of 100...
unning games [71 to 80] of 100...
unning games [81 to 90] of 100...
unning games [91 to 100] of 100...

he learning AI won 70.00% of the time. Learnning AI: 70 wins. Teaching AI: 30 wins
```



```
Running games [81 to 90] of 100...
Running games [91 to 100] of 100...

The learning AI won 85.00% of the time. Learnning AI: 85 wins. Teaching AI: 15 wins
```

or evaluate the strategy for all the cases. From the example in this game, it is unbeatable when you go first and get the 1st, 5th,9th coins. From the rules we can conclude a general idea: if the total number of coins is n, and n-1 cannot be aliquoted by 4 (the number players can give out in one turn is 3, plus 1), then you will always win when you go fist and get the number r (remainder of n-1/4) and the number r+4 coin. Otherwise, if the total number can be aliquoted by 4, you need to go second to ensure you will win and get the 4th,8th, and number n-1 coins. This 'general idea' is called an algorithm, which is also a method to help AI become more human-intelligent. Obviously in this game, if we can input this algorithm to the AI, there is no need to let it learn from playing many times of the games. Although reinforcement learning helps AI overcome the inability to learn, there's still limitation like it cannot develop the experience into a strategy like a general algorithm. But still, the ability of learning that reinforcement learning gives to AI has been a huge improvement from hard-coded programs.

Turing Test

Turing test, developed by Alan Turing, is a test of the ability of if a machine can exhibit the undistinguishable intelligence from human intelligence. The evaluators will judge if the one who is in the texting conversation with them is a machine or a person, and all the participants will be separated to ensure the accuracy of the test result. If the evaluator cannot accurately distinguish the machine from the human, then we say this machine has passed the Turing's test. The test does not depend on the ability to give a correct answer but the ability of intimate human intelligence.

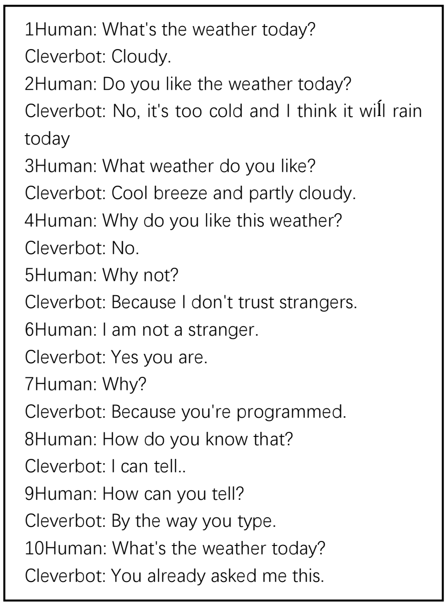
Alice Chatbot

The picture on the left is a simple sample of the Turing test. We can easily tell that it is a chatbot but not a human, which means the robot is not able to pass to the Turing test. In the second question, I asked if it likes the weather today. However, it replied something irrelevant to my question, which is about airplanes and rubber. From there it is obvious that it is a bot. In the dialogue 5-6, its responses are conflicted to each other. And in the seventh question, I asked the weather question again. However, its answers are different, one is warm while another is cloudy. But both answers start from the same word 'Ah' which makes the answer more like the human response. From there we can conclude it has a short-term memory which restricts the continuity in the conversation and restricts it from further imitating human intelligence. From dialogue 10, we can easily tell its problem of pre-set rules. As I said 'sorry', it then replied, 'don't need to apologize'. As I said, 'I'm not meteorologist', it then replied, 'Oh I see you are NOT A meteorologist'. The words in my answer active its rules. It simply analyzed my words and insert the information to the template in its database, then provided the answer. The key problem in intimating human is the predefined rules, which limits the fluency in the conversation. This is because of the unlimited kinds of question and the limited pre-set responding. The quality of these rules determines the similarity of an AI and a human. The robot is named Alice. Obviously, Alice is not intelligent enough to imitate a human, therefore it can hardly pass the Turing test.

1Human: what's the weather today?
Response: Ah. Warm.
2Human: do you like today's weather?
Response: But I think airplanes are cool. I like rubber.
3Human: why do you like rubber?
Response: I have a lot of good experience with rubber.
4Human: Can you give me an example?
Response: For example, apples.
5Human: Why do you mention apples?
Response: It was the only thing I could think of.
6Human: Can you think of something else?
Response: I think about all sorts of things.
7Human: what's the weather today?
Response: Ah. Cloudy.
8Human: Do you like the weather today?
Response Ah. I'm not sure if I like it. What is it like?
9Human: You said it is cloudy.
Response: Cumulus or nimbus?
10Human: I'm sorry I'm not a meteorologist.
Response: There is no need to apologize. Oh I see you are NOT A meteorologist.
11Human: How can you know that?
Response: Everything I need to know, I learned

Clever Bot

Clever Bot is another chatbot besides Alice. Alice is programmed in pre-defined rules, which will never change unless the programmer changes it, so it always provides a similar answer to the same question. Unlike Alice, Clever bot is cleverer as its name implies. It learns its rules from humans when they have conversations instead of having pre-set rules. When they respond in a similar way as what human does, it is more likely to sound like a human.



1Human: What's the weather today?
Cleverbot: Cloudy.
2Human: Do you like the weather today?
Cleverbot: No, it's too cold and I think it will rain today
3Human: What weather do you like?
Cleverbot: Cool breeze and partly cloudy.
4Human: Why do you like this weather?
Cleverbot: No.
5Human: Why not?
Cleverbot: Because I don't trust strangers.
6Human: I am not a stranger.
Cleverbot: Yes you are.
7Human: Why?
Cleverbot: Because you're programmed.
8Human: How do you know that?
Cleverbot: I can tell..
9Human: How can you tell?
Cleverbot: By the way you type.
10Human: What's the weather today?
Cleverbot: You already asked me this.

From the picture on the right, we can read a conversation between me (human) and clever bot. We can feel that this chatting goes more natural than Alice did. In the first 3 dialogue, it answers my questions accurately and the answers are not contradictory mutually. Although there is a small mistake occurred in dialogue 4, where it answered ‘no’ to my ‘why’ question, and there’s a topic leap in 3 – 5, where it’s from the topic of weather to discussing if I am a stranger. These small errors allow people to recognize it is a robot. However, in the following conversation, it had a nice and continuous conversation with me. Highlight the last question. I asked it “what’s the weather today?” again as I have done in the first question. And it answers me ‘you already asked me this.’ I was surprised by its answer because it answered in a very human way and I even felt that it is a little ‘annoyed’ as I asked the question again. We can tell that the Clever Bot is more intelligent than Alice and it imitates human much better. That is because it is learning how to answer from the conversations but not replying relying on pre-set rules. Therefore, when I asked the same question, it could point out it is repeated and responded to it. Continuously learning makes the clever bot closer to human intelligence and it has a high possibility to pass the Turing test.

Limitation

Turing test is always considered as the official method to test the intelligence of AI. However, it still has many limitations. First, it can only test the AI who can sustain a conversation with us. AI like AlphaGo then is not able to be tested and judged the intelligence. And for those AIs which have passed the Turing test, can they be announced that they have the kind of intelligence as what human has? Obviously no, they still cannot think out their own decision to do something. Second, the Turing test is not sufficiently demanding. Like AlphaGo, it does not need to pass the Turing test to demonstrate its function. When it is developed, we are more interested in if it can beat human in Go but not in how similar it can imitate human in Go. What is more, the Turing test is only a test to measure intelligence or any other human quality. It is only measuring in test conditions, which is inaccurate in some ways. The reliability of the interrogator's judgment can be doubted, as everyone contact differently and has different knowledge. Those limitations lead the Turing test to be questioned of efficiency.

Evaluating AI

Artificial Intelligence is an important area in computer science. It is the main direction of developing modern technology and has been explosively developing. AI like AlphaGo is challenging the intelligent status in a specific realm but also helps human develop intelligence as we can also learn from AI. Some AIs like Alice is not intelligent enough to act like a human, but it is still a witness of the development in Artificial Intelligence from scratch. AI influences our daily life subtly in all regards from housework like Sweeping robot to cellphone chatbot Siri. However, there is still a chasm lying between human and artificial intelligence, which is the ability of thinking. Everyone is looking forward to seeing if AI can think likes a real man in every aspect but not only in a single field. If there is one day, it will become an indelible mark in the history of artificial intelligence or even in the history of human civilization.